REAL TIME VIDEO GENERATION

API Documentation

Version 5.0.3.6

TABLE OF CONTENTS

1. OVERVIEW

1.1. Introduction

Idomoo is the leader in personalized video, enabling customer communication and direct marketing using automatically generated video.

A short video can effectively communicate a complex message, create emotions and drive customers to action. Whether your goal is customer satisfaction, sales or cost reduction, video is the most effective way to personalize communication with impact.

Combining quality video with personalization increases the visibility and appeal of your message, maximizes customer engagement, and improves responsiveness.

Idomoo's Video as a Service (VaaS) platform receives customer data from standard CRM systems or databases and generates millions of tailor-made, high quality, professional video clips.

By automatically integrating personal customer information within compelling videos, and distributing these videos via a range of convenient channels from email to web to mobile, Idomoo enables companies to communicate in a fun and personal manner.

Following successful deployments with customers worldwide, Idomoo has proven that its solution lowers service costs, improves customer satisfaction and increases average revenue per customer.

1.2. Purpose

This document depicts the workflow of operating Idomoo's video platform API for generating On-The-Fly or On-Demand Video content.

2. API INTERFACE

The following paragraphs describe the various aspects involved in interfacing with Idomoo's API.

2.1. API End-Points

Idomoo's API currently hosts several End-Points.
Choose the API end point to ensure secured and faster data transfer

Environment	FQDN
Production – EU (Europe)	https://eur-api.idomoo.com/ext/cg
Production – US (United States)	https://us-api.idomoo.com/ext/cg

The Production End-Points will be used to serve functional storyboards only

2.2. API Headers

Key: Content-Type, Value: application/json

2.3. API Video Generation Round Trip Test

In order to test the integration of your/your client's platform with Idomoo's API prior to having your video storyboard ready on Idomoo's platform, you can generate test videos with Idomoo's example storyboard that is available under your account.
Implementing API calls must be done using the Request Signing mechanism, as described here

3. API REQUEST SPECIFICATION

3.1. Clip Generation Action

This action is used in order to trigger video generation according to data passed in the request itself. Data posted is composed of the following sections:

3.1.1 Action HTTP Path

Please refer to API End-Points

3.1.2 Action Request Formats

There are several formats to send via POST request to Idomoo's API. The platform will differentiate between these formats according to the “Content-Type” HTTP header set in the request. Failure to set the “Content-Type” in accordance with the table below will result in a request validation error:

Content	Content-Type Header	POST Body Example (Formatted)
JSON Document	application/json	{ "response_format":"json", "video":{ "storyboard_id":123, "account_id":234, "landing_page_id":345, "storage_id":1, "statistic_id":"Testing_API", "video_file_name":"0123456789abcdefghijklmnopqrstuvwxyz12", "authentication _token":"2378nasdn7834d9m80e9sdan098m-87b", "output_formats":[{ "format":"VIDEO_MP4_V_X264_640X360_1600_A_AAC_128" }, { "format":"IMAGE_JPEG_640X360", "marker":[{ "time":"00:00:04.000", "index":1 }] }], "data":[{ "key":"A", "val":"1" }] } }

3.1.3 Action Response Formats

Each request sent to Idomoo's API has its response format set according to the value of the “rf” parameter, regardless of which input format was chosen by the request originator. The following table describes the various output formats provided by Idomoo's API:

"response_format" Parameter value	Returned content	Example of successful response	Failure response format	Example of failure response
json	A JSON formatted document containing generation information regarding the requested video/s	*{ "status":"GENERATION_SUCCEEDED", "video":{ "output_formats":[{ "format":"VIDEO_MP4_V_X264_640X360_1600_A_AAC_128", "duration":"00:00:30.000", "cost":"6.0", "links":[{ "url":"https://videos.idomoo.com/234/Testing_API/0123456789abcdefghijklmnopqrstuvwxyz12.mp4", "landing_page_url":"https://play.idomoo.com/videos/?url= https://videos.idomoo.com/234/Testing_API/0123456789abcdefghijklmnopqrstuvwxyz12.mp4", "unique_id":"234/Testing_API/0123456789abcdefghijklmnopqrstuvwxyz12" }, { "format":"IMAGE_JPEG_640X360", "duration":"00:00:01.000", "cost":"3.0", "links":[{ "url":" https://videos.idomoo.com/234/Testing_API/0123456789abcdefghijklmnopqrstuvwxyz12.jpg", "landing_page_url":"", "unique_id":"234/Testing_API/0123456789abcdefghijklmnopqrstuvwxyz12" }] }] }], "internal_id":443, "internal_code":0, "error_description":"", "total_cost":"6.0" } }*	json	*{ "status": " GENERATION_FAILED", "internal_id":443, "internal_code":564654, "error_description":"" }* OR *{ "status": "GENERATION_FAILED", " error_description ": "User Account ID : '234', is not active!" }*

Note : Video and Thumbnail links are https and can't be changed to http.

3.1.4 Action parameters

3.1.4.1 Request Scope parameters

This section refers to parameters present in the scope of the entire request

API Parameter Name	Description	Possible Values	Default Value	Mandatory	Dependencies/ Restrictions
Response format	Determines the formatting of which the response of video generating will be return	json See “Response Table” below for detailed description of the each possible value	N/A	Yes	POST requests
video	Structured video Generation Data	Structured data passed in accordance to matching Input format. **JSON Example:** *{ "video":{ "storyboard_id":123, "account_id":234, "landing_page_id":345, "storage_id":1, "statistic_id":"Testing_API", "video_file_name":"0123456789abcdefghijklmnopqrstuvwxyz12", "authentication_token":"2378nasdn7834d9m80e9sdan098m-87b", "output_formats":[{ "format":"VIDEO_MP4_V_X264_640X360_1600_A_AAC_128" }, { "format":"IMAGE_JPEG_640X360", "marker":[{ "time":"00:00:04.000", "index":1 }] }], "data":[{ "key":"A", "val":"1" }] } }*	N/A	Yes	

3.1.4.2 Video Scope parameters

This section refers to parameters present in the scope of the video. single request can generate a few formats.

API Param Name	Description	Possible Values	Default Value	Mandatory	Dependencies/ Restrictions
storyboard_id	Storyboard ID, as given by idomoo's support team or via the storybuilder suite	Number	N/A	Yes	Account_id must be the owner of the storyboard_id
account_id	Account ID, as given by idomoo's support team or via the storybuilder suite	Number	N/A	Yes	Account_id must be the owner of the storyboard_id
landing_page_id	Landing page ID, as given by idomoo's support team or via the storybuilder suite	Number	N/A	Yes	Account_id must be the owner of the landing_page_id
storage_id	Storage ID, as given by idomoo's support team or via the storybuilder suite	Number	N/A	No	As default the video are stored on Idomoo's storage
statistic_id	This Identifier will be used to detect the call in storybuilder suite analytics. Omitting this parameter the statistic ID will be signed as '"0000"	Set of 0-9, A-Z characters. Example: april2015	N/A	No	ID given will be used "as-is"
video_file_name	This Identifier will be used as part of the URL identifier of the generated video. Omitting this parameter will cause a self-generated "hashed" external ID to be used	Set of 0-9, A-Z characters, Up to 128 characters	N/A	No	ID given will be used "as-is", with no additional encryption/Hashing
authentication_token	Request Hash String, generated as described here	The string received as the result of the request signing function, as described here	N/A	Yes	
output_formats	Structured array of video / thumbnail generating formats	Structured data passed in accordance to matching Input format **JSON Example:** *"output_formats":[* *{* *"format":"VIDEO_MP4_V_X264_640X360_1600_A_AAC_128"* *},* *{* *"format":"IMAGE_JPEG_640X360",* *"marker":[* *{*	N/A	Yes	

		"time":"00:00:04.000", *"index":1* *}* *]* *}* *]*			
format	Requested format of the API request (Video,image , etc.)	See possible formats below	N/A	Yes	
marker	Structured array of time / index That descries how to capture the thumbnail	Structured data passed in accordance to matching Input format. **JSON Example:** *"marker":[* *{* *"time":"00:00:04.000"* *, "index":1* *}* *]*	N/A	Yes for thumbnail format	
time	Second to capture the thumbnail	Time format HH:mm:ss.SSS	N/A	Yes for thumbnail format	Time format must be HH:mm:ss:SSS
index	Indication number of thumbnails sent	Number	N/A	No	Index will included as postfix to the thumbnail UR L
data	Structured array/collection of Key/Value pairs holding all of the required data needed to generate the video	Structured data passed in accordance to matching Input format: **JSON Example:** *"data":[* *{* *"key":"A",* *"val":"1"* *},* *{* *"key":"B",* *"val":"2"* *}* *]*	N/A	Yes	Values encoding must be in UTF-8 encoding

4. OUTPUT FORMAT SPECIFICATION

Idomoo's video generation API supports various video and image formats to be passed upon posting the video generation request.

4.1. Technical definitions

- Bit-rate - The maximum number of bits that can be used to represent data in each single second of the video. Choosing a high value for this number can drastically improve both visual and audible content quality, at the price of increasing storage volume and bandwidth required to store and stream each video. Idomoo's API currently supports a pre-defined set of values for both video and audio bit rate , based on best practices and experiences

- Video Dimensions – The Width and Height of the Video Stream, expressed in <WIDTH>x<HEIGHT>. Idomoo supports both SD (640x360) and HD (1280x720)

- Video Codec - Encoding Library used to generate the visual content of the video(i.e. H264)

- Audio Codec - Encoding Library used to generate the audible content of the video(i.e. AAC)

- Video/Image Container – The Container used to present the encoded video and audio streams(i.e. MP4 , HLS)

4.2. Technical Specification

The table below specifies the Output formats recognized by Idomoo's API:

API ID	Container	Video Codec	Video Bit Rate (Kbps)	Video Dimensions	Audio Codec	Audio Bit Rate (Kbps)
VIDEO_HLS_V_X264_640X360_800_A_AAC_128	HLS	H264	800	SD(640x360)	AAC	128
VIDEO_HLS_V_X264_640X360_1600_A_AAC_128	HLS	H264	1600	SD(640x360)	AAC	128
VIDEO_HLS_V_X264_1280X720_800_A_AAC_128	HLS	H264	800	HD(1280x720)	AAC	128
VIDEO_HLS_V_X264_1280X720_1600_A_AAC_128	HLS	H264	1600	HD(1280x720)	AAC	128
VIDEO_HLS_V_X264_1280X720_4000_A_AAC_128	HLS	H264	4000	HD(1280x720)	AAC	128
VIDEO_MP4_V_X264_640X360_800_A_AAC_128	MP4	H264	800	SD(640x360)	AAC	128
VIDEO_MP4_V_X264_640X360_1600_A_AAC_128	MP4	H264	1600	SD(640x360)	AAC	128
VIDEO_MP4_V_X264_1280X720_800_A_AAC_128	MP4	H264	800	HD(1280x720)	AAC	128
VIDEO_MP4_V_X264_1280X720_1600_A_AAC_128	MP4	H264	1600	HD(1280x720)	AAC	128
VIDEO_MP4_V_X264_1280X720_4000_A_AAC_128	MP4	H264	4000	HD(1280x720)	AAC	128
IMAGE_JPEG_640X360	JPEG	N/A	N/A	SD(640x360)	N/A	N/A
IMAGE_JPEG_1280X720	JPEG	N/A	N/A	HD(1280x720)	N/A	N/A

Recommended Output formats for HLS:

- Videos with 15 fps:

  - SD: VIDEO_HLS_V_X264_640X360_800_03_g45
  - HD: VIDEO_HLS_V_X264_1280X720_1600_03_g45

- Videos with other fps:

  - SD: VIDEO_HLS_V_X264_640X360_800_03_g75
  - HD: VIDEO_HLS_V_X264_1280X720_1600_03_g75

4.3. Technical Restrictions

- JPEG container can be used only with mp4 container request

5. API IMPLEMENTATION GUIDELINES

Idomoo's API calls are done using standard HTTP (RESTFul) protocol mechanics, and thus, can be implemented using a variety of programming languages & frameworks. Regardless of choice of technology, several guidelines must be followed in order to ensure the validity of both API and Analytics data that's provided back to the customer by Idomoo:

5.1. Request Signing Mechanism

In order to ensure the ID of the remote entity requesting the video generation and avoid an abuse of both Idomoo's API platform and the customer's Account, a signature mechanism must be applied to all video generation requests.
The result of that signature mechanism is in the form of a hash, and applied to request it, using the parameter "authentication_token", as described below.

5.1.1 Request Signing Mechanism Input

The Signature mechanism receives an input string composed of the following data:

- Both "key" and "val" field values of the elements, concatenated without delimiters and according to order of their appearance in the API Call
- Each account has a secret key for API usage in his account setting area. It can be re-generated by the user himself from the account setting area.

For Example, if the request ***"data"*** fields contains the following value: ***[{"key": "A", "val": "1"}, { "key" : "B" , "val" : "2"}]*** , and the secret key is ***SHAREDSECRET*** , the string to be passed for request signing will be – ***A1B2SHAREDSECRET***

Note: The input string to the mechanism must have the same order of key value as it sent through the JSON request. Different orders will generate different authentication tokens.

5.1.2 Request Signing Mechanism Implementation

The Signature Mechanism use standard techniques in order to create a matching hash. Following are code samples in several languages/frameworks:

5.1.2.1 PHP

```
function   signRequest($str){
   $str   =  str_replace('\n', "\n", $str);
   $str   =  mb_convert_encoding($str, 'ISO-8859-1', 'UTF-8');
   $md5   =  md5($str,true);
   $hash  =  base64_encode($md5);
   $hash  =  str_replace('+','.',$hash);
   $hash  =  str_replace('/','_',$hash);
   $hash  =  str_replace('=','-',$hash);
   return $hash;
}
```

5.1.2.2 Java

```
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import org.apache.commons.codec.binary.Base64;

public String signRequest(String stringToSign) throws CloneNotSupportedException,
UnsupportedEncodingException, NoSuchAlgorithmException {

    String hash;

    MessageDigest md = MessageDigest.getInstance("MD5");
    MessageDigest tc1 = (MessageDigest) md.clone();

    byte[] toChapter1Digest = tc1.digest(stringToSign.toString().getBytes( "ISO-8859-1"));

    byte[] b = Base64.encodeBase64(toChapter1Digest);
    hash = new String(b,  "ISO-8859-1");
    return hash.replace('+', '.').replace('/', '_').replace('=', '-');
  }
```

5.1.2.3 C#

```
public static string signRequest(string str) {

        byte[] bytes =
        MD5.Create().ComputeHash(Encoding.GetEncoding(28591).GetBytes(str));
         str = System.Convert.ToBase64String(bytes);
         str = str.Replace('+', '.').Replace('/', '_').Replace('=', '-');
         return str;
      }
```