API v3 Documentation

Table of Contents

AlphaPicture GmbH
Hauptstr. 64
91054 Erlangen, Germany
Phone: +49 9131 9411846
support@alphapicture.com

1 Introduction

AlphaPicture offers an API to retrieve personalized images. Such images require preparation in order to be personalized and once this preparation has been performed by us we call those images "Motifs". Those motifs are then loaded into our personalization software which will, after receiving text input, render personalized images.

This document describes the third version of the API which is designed to scale virtually indefinitely through the use of cloud computing. Small jobs are still run on continuously available servers in order to achieve quick processing time.

2 API Description

The third version AlphaPicture API is just as previous versions HTTP-based. But while previous versions were limited to requesting one image per HTTP request, the new API can now return many images of one or even multiple motifs in one go.

The API is REST-based and is setup to use JSON for input and output. If you have a specific need to add another protocol and can motivate why it should be there please contact us.

The next sub-sections will explain functionalities which are relevant for implementation. More features will be exposed in the near future.

2.1 Authentication

The API requires authentication in order to prevent malicious requests. Never, ever, should you expose any AlphaPicture API directly to your customers! In all cases the only program making requests to the API should be fully in your control and not exposing any functions to the outside world.

Authentication happens through three HTTP-headers:

Header	Contents	Used in
APIV3-Account	Your API username as provided by AlphaPicture	Motifs/Job/JobInfo/Image
APIV3-Password	Your API password as provided by AlphaPicture	Motifs/Job/JobInfo/Image
APIV3-Signature	A SHA512 hash of the request body, then concatenated with your secret key provided by AlphaPicture and this concatenation run through SHA512 again	Job/JobInfo/Image

While this may seem cumbersome, this setup prevents third parties making unauthorized requests which may directly incur costs. Password is used in places where no request body is available.

2.2 Functions

2.2.1 Motifs

The "Motifs" functions exposes information about all motifs that are available to your API account. This data consists out of some basic data such as name, maximum resolution but also contains information about the typefaces used. The latter is important for performing data testing. We strongly suggest you save this motif information locally and periodically update it through use of the "lastmodified" key.

2.2.1.1 Motifs - Request

Type	GET
URL	http://v3.alphapicture.com/Motifs
Authentication	APIV3-Account and APIV3-Password
Body	None

2.2.1.2 Motifs - Response

The response will be a JSON array containing information about all accessible motifs. A single motif will be used for illustration below

```
{
             "id" : 18, // This is the motif number
             "version" : 14, // Motif version
             "name" : "Gingerbread Heart", // Motif name
             "creationdate" : "2002-10-22T00:00:00.000Z", // Creation date
             "lastmodified" : "2011-11-21T00:00:00.000Z", // Last modification date
             "alternatives" : [{ // This is a list of all alternatives existing for this
motif, most                                  // motifs have multiple alternatives with
slight (or big) differences
                          "alternative_id" : 1,     // alternative number
                          "motif_id" : 18,    // motif id
                          "name" : null,      // alternative description, null if only
one                                                    // alternative is
present
                          "photographer" : "Michael von Aichberger",
                          "programmer" : "Michael von Aichberger",
                          "client" : "AlphaPicture",// This will be AlphaPicture for
                                                    // 'public' motifs
                          "programmers_note" : null,// Some motifs have text here which
```

```
will give some                                                    // guidelines how
to format text
                        "sample_text" : {   // This can be used to create sample images
                               "1" : ["Merry Christmas"],
                               "2" : ["with"],
                               "3" : ["AlphaPicture"],
                               "4" : ["xxx"]
                        },
                        "lines" : {  // This is an array with all lines available in
this alternative
                               "1" : {
                                      "typeface" : "1",   // Each line has a typeface
associated with it
                                      "length" : "3650"   // Each line has a certain
maximum length
                               },
                               "2" : {
                                      "typeface" : "1",
                                      "length" : "3810"
                               },
                               "3" : {
                                      "typeface" : "1",
                                      "length" : "2800"
                               },
                               "4" : {
                                      "typeface" : "1",
                                      "length" : "1850"
                               }
                        },
                        "original_rect" : "0, 0, 3068, 4095",   // Original size of the
alternative
                        "bounding_rect" : "958, 1953, 2568, 3810"     //
Personalization area
                    }
             ],
             "typefaces" : [{    // Each motif has a set of typefaces associated with it
                        "typeface_id" : 1,  // Typeface have an id, this is referenced
by lines in                                          //alternatives
                        "motif_id" : 18,
                        "typewidthL" :      // This is a list of how much space every
character takes,                                          // refer to data
testing section for how-to-use
"[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,129,99,0,0,0,0,231,99,0,0
,0,247,112,175,99,0,260,223,249,211,226,240,235,255,240,220,104,0,0,0,0,191,233,340,227,2
34,320,237,292,292,217,246,298,231,266,375,311,238,237,255,244,240,255,254,238,287,234,23
8,240,0,0,0,0,0,0,0,212,202,194,249,213,232,241,247,178,220,220,216,252,216,188,249,244,237
,231,228,223,196,291,181,275,275,.....]",
                        "symbolL" : null,   // Refer to data testing section
                        "charspacing" : "0.0000"  // Refer to data testing section
                    }
             ]
      }
```

2.2.2 Image

The "Image" function allows to generate single images rapidly and retrieve them instantly. So 1 request results in 1 rendered image.

2.2.2.1 Image Request

Type	POST
URL	http://v3.alphapicture.com/Image
Authentication	APIV3-Account and APIV3-Signature

Body	<pre>{
	"MotifId": 614, // Id of the motif
	"AlternativeId": 2, // Alternative Id
	"Lines": { // Array of lines
	"1": "JOHN DOE"
	},
	"Dimensions": { // Dimensions in pixels
	"w": 881,
	"h": 327
	},
	"SourceRect": { // Optional: SourceRect is for retrieving a certain cut-out of
	// the original image
	"x1" : 1532,
	"y1" : 1353,
	"x2" : 3801,
	"y2" : 2195
	},
	"Watermark": false // Optional: set to true if image should be watermarked
	}</pre>

2.2.2.2 Image Response

Response will be a single JPEG-image

2.2.3 Job

The "Job" function allows to generate large amounts of images.

2.2.3.1 Job Request

Type	POST
URL	/Job
Authentication	APIV3-Account and APIV3-Signature

Body	<pre>{
	"Images" : [{ // Array of motifs that need to be rendered
	"MotifId" : 4,
	"AlternativeId" : 1,
	"Template" : { // See below for explanation on template</pre>

```
                                '1+2' : "Hello %firstname% %lastname%"
                        },
                        "Filename" : "One_%number%", // Filename, without
JPG extension
                        "Dimensions" : { // Dimensions in pixels
                                "w" : 1000,
                                "h" : 750
                        },
                        "Watermark" : false
                }
        ],
        "Callback" : "http://some.callback.com/", // HTTP callback
called when job done
        "OutputOptions" : { // This are options related to the output
of this job
                "OutputMethod" : "HOTLINK" // See below for options
        },
        "Data" : [{ // This is the data object which used in
substitution, see below
                        "firstname" : "John",
                        "lastname": "Doe",
                        "number" : 1
                }, {
                        "firstname" : "John",
                        "lastname": "with a long name",
                        "number" : 2
                }
        ]
};
```

The body outlined above is a simplified example of what is possible in a job request. Some important parameters are explained in more detail below:

Parameters explanation

Template	Images in a job are rendered based on a template. A template is an object which defines the personalized text that is rendered in an image. A motif can feature one or more personalizeable lines of text. A text can spread over a single line or over multiple lines. If multiple line numbers are passed (in a '1+2+3' fashion) then text will be split on spaces and/or ampersands while trying to fill the first lines as much as possible. The text in an image can be static or variable (personalized, derived from a database) or a mix of both. For the personalized part, variables are used. Variables are noted with percent signs and refer to data objects which are passed in the data array. Naming of these variables is completely arbitrary – except that variable names themselves cannot contain percent signs.
Filename	Filenames use the same variables as templates
Callback	When a job is done and if a callback URL has been passed this callback will be called with a JSON body. The fields JobId, Status, Key and DownloadURL will be passed back to this callback
OutputMethod	Currently we have two output options: - HOTLINK → this will output all images on a CDN (Content Distribution Network) for direct hotlinking usage. The hotlink URL will be available in JobInfo. - ZIP → the files will be zipped up on our servers and made available for downloading
Data	Jobs work based on data-objects, an array of data objects should be passed in. For every data

	object one image is rendered for every object in the "Images" array. Thus passing 5 different motifs with different templates and sizes and 1 data object will render these 5 motifs for this single data object. Data objects can contain any key/value

2.2.3.2 Jobs Response

A JSON object in the following format will be returned:

```
{
	JobId : 1234,
	Error : false,
	Status : 'IN_PROGRESS',
	Key : 'f81d4fae-7dec-11d0-a765-00a0c91e6bf6',
	CDN : 'http://cdn.alphapicture.com/f81d4fae-7dec-11d0-a765-00a0c91e6bf6/'
}
```

Good practice is to save JobId along with the request that you have stored somewhere, this id can later be used to fetch the status of a job and retrieve other information about it. CDN links are not available for jobs which do not have the "HOTLINK" OutputType.

2.2.4 JobInfo

The "JobInfo" function allows to retrieve information such as status about a specific job.

2.2.4.1 JobInfo Request

Type	POST
URL	http://v3.alphapicture.com/JobInfo
Authentication	APIV3-Account and APIV3-Signature

Body	{
	"JobId": 1234
	};

2.2.4.2 JobInfo Response

A JSON object in the following format will be returned:

```
{
	JobId : 1234,
	Error : false,
	Status : 'IN_PROGRESS',
	Key : 'f81d4fae-7dec-11d0-a765-00a0c91e6bf6',
	CDN : 'http://cdn.alphapicture.com/f81d4fae-7dec-11d0-a765-00a0c91e6bf6/'
```

```
}
```

Good practice is to save JobId along with the request that you have stored somewhere, this id can later be used to fetch the status of a job and retrieve other information about it. CDN links are not available for jobs which do not have the 'HOTLINK' OutputType

2.2.5 Result

The "Result" function allows to retrieve resulting images or a zipfile from a specific job.

2.2.5.1 Request

Type	GET
URL	http://v3.alphapicture.com/Result/{key}(/{filename})
Authentication	None
Body	None

2.2.5.2 Response

Requesting with just a key (see JobInfo or response from Job for retrieving a key) will result in downloading a zipfile. If a filename is specified then this specific file is returned as JPEG.

3 Data testing

One important part of integrating the AlphaPicture API is understanding that not all text will work in any motif. There are constraints on a number of things:

- Characters available in a motif
  Imagine fonts that were hand-made which might not have all eastern European characters
- Length of text in a line
  In order to prevent text becoming unreadable or looking unrealistic we enforce constraints on the maximum length a line can be

We always do a server-side check of all these constraints and will notify your system when an impossible request is encountered, but in order to deliver end-users a smooth experience we strongly advise you to implement our data testing algorithm in any web front-end for end-users.

3.1 Step 1 – Checking for character availability and possible substitution

In order to check whether a character is available within a certain motif you should follow the following procedure:

- Identify the Unicode position of the character you want to verify (i.e. "€" = Unicode 128)
- As any line in a motif can use another typeface, identify which typeface the line this character should be placed in is using. This can be done by looking at the "lines" definition in the motif information retrieved from the API. There you will have the typeface-to-line allocation.
- Within any typeface definition there is a field called “typewidthL” which contains a rather large array. This array holds the virtual character widths of all characters supported by this typeface. We will need those values for the calculation of the total text width. Note that in this document, "width" and "length" of a text are used interchangeably, always referring to the width of a text, because we do not just count the number of letters in a line. Instead, we add the individual widths of the characters plus a value for the space between characters plus in some cases additional calculations for kerning. For example: If you want to look up character 128 then check the 128th position in this list. If there is a number greater than zero this means the character is supported, if there is a width of 0 then this character is not supported. Note that the indexing of that array starts with the value of 1, not with 0 as in many programming languages. Thus, the index is indentical to the unicode of a character.
- If a character is not supported then you should check if it has viable alternatives available (optional) by checking the “substitution_list.json” file. This contains a field for every character that can be substituted which holds an array of possible substitutions.

  For example, if a typeface supports only uppercase letters and you pass lowercase text, you don't want to the motif not to render any text. Instead you expect your lowercase text to be converted into uppercase text. Or, if a motif doesn't support characters with accents, like "é" you want those to be converted to the base characters without accent. In substitution list there are even more comlicated suggested substitutions. For example: your text contains a lowercase o umlaut "ö", it might be converted to "oe". If "oe" is not possible (in the case of the typeface being uppercase only), "Ö" is tried. And if that's not possible either, "OE" is tried. So if there's a substitute in the list, you should very that the suggested character is available in typewidthL as above. You might do this as long as you have either found a viable substitution, or until there is not suggestion for substitution left. If you found a substitution, everything will be ok, otherwise it is not possible to render this motif with that specific text.

3.2 Step 2 – Checking if width has been exceeded

Once you have identified the width of all characters on a line, and performed possible substitution it is time to add up all the values from typewidthL and check against the "length" property of the specific line. You should however add a variable amount of length according to the number of characters if charspacing is not set to 0.000 for the typeface applicable to the current line. If the total length of your line is less than the allowed maximum you are ready to render images.

4 Code samples

In order to use the included code samples you will have to install Node.js - if it is not already existing on your system. See http://nodejs.org/ for further reference.

You should unzip the samples to a folder and then from within this folder run "npm install request" in order to fetch a dependency. Then you can run the samples by running "node 1_motifs.js" from the command line. You will however need to enter your username/password/secret in each sample file for them to run correctly.

4.1 Retrieve motifs and save to a local database

This sample shows how to query the AlphaPicture API for all motifs available to you, you should this somewhere locally.

4.2 Retrieve a single image

This sample retrieves a single image and stores it in the output folder with a prefix of 1_, feel free to adjust parameters. This functionality should be used if you want to quickly show a preview of a single image.

4.3 Retrieve a single image for all available motifs (gallery)

This sample uses the database saved by sample 1 in order to fetch all files with their stored sample text.

4.4 Retrieve a job with 100 names for 1 motif

This sample retrieves a single motif with a dataset of 100 names.

4.5 Retrieve a job with 10 names for 10 motifs

This sample retrieves 10 motifs with 10 different data objects.

4.6 Job with callback

This sample runs a dummy job with a callback attached to requestb.in, you should set up your own requestb.in or callback URL in order to see how this is working.

5 Various mentions/notes

- When making requests towards the API always be sure to include a Content-Type: application/json header
- This is a first public release of the API, if you encounter any problems, please do let us know!
- If you have any feature requests and/or improvement suggestions then please communicate them to m.v.aichberger@alphapicture.com
- If you require any support please contact k.faro@alphapicture.com