Partner Api - Dokumentation

Getting started

You can get your API key right away. Just sign up at pokamax.com if your haven't already, sign in to your new account and fill out the reseller form located here: https://pokamax.com/us/account/api / https://pokamax.com/de/konto/api

After that you can get your API key in your account. Just klick on "My Account" -> "API" and there it is. The new API key is already active and in testing mode. Testing mode means, that you can transmit your orders over the api but they get canceled right away. So you won't geht charged for testorders.

You are now able to implement your part while we arrange the prices and conditions.

When both sides aggree to the prices and conditions, Pokamax approves your api key for going live. You will get a notification and a contract form with the prices and conditions which you have to accept. From this moment on your API will be live and all transfered orders will be printed and charged.

Billing

We will send you a monthly invoice.

Technical Description

API Location

All API endpoints are located under `https://pokamax.com/apis/reseller/v1/`. So if below the *"**orders**" endpoint* is mentioned then you can find it at `https://pokamax.com/apis/reseller/v1/`**`orders`**

Authorization

Every request should contain an **Authorization** header with the api credentials like this:

```
Authorization: Token token="abc123", email="reseller@example.com"
```

The token is your api key and the email is your accounts email-address.

Ordering

The request has to be a **POST** request on the ***orders** endpoint.*

You can transfer one or more cards per order and one order per request. Your cards have to be ordered in a numbered array like this:

```
order[cards_attributes][0][product_name]=...
order[cards_attributes][0][name]=...
...
order[cards_attributes][1][product_name]=...
order[cards_attributes][1][name]=...
...
```

You can decide if you want render the cards yourself (advanced) or by pokamax (easy).

Rendered by Pokamax

If you want to start right away without the hassle of generating the backside of your cards yourself, you leave that to us. Just send us the text, address and a front picture and the card is complete.

Group	Parameter	Type	Default	Mandatory	Possible values
	product_name	text		*	"standard","panorama","jumbo",...
	ship_date	date			"2015-12-24"
Frontside	picture	file		just use one of them	
	picture_id	int			ownpicture id or gallerypicture id
Address	company	text			
	title	text			
	name	text		*	
	street	text		*	
	zip	text		*	
	city	text		*	
	state	text			
	country	text		*	"DE","US",... ISO2 Standard
Backside	text	text			
	text_color	colorcode	#000000		#000000-#ffffff

Example

```
order[cards_attributes][0][product_name]=standard
order[cards_attributes][0][picture]=@/front.jpg
order[cards_attributes][0][name]=Max Mustermann
order[cards_attributes][0][street]=Musterstr. 22
order[cards_attributes][0][zip]=22345
order[cards_attributes][0][city]=Berlin
order[cards_attributes][0][country]=DE
order[cards_attributes][0][text]="This is a greeting text.\nSecond line..."
order[cards_attributes][0][text_color]=#000000
=> Status: 201, Body:
{"id":1950418,"cards":[{"id":1950777,"company":null,"title":null,"name":"Tester","stre
et":null,"zip":null,"city":null,"state":null,"country":"Germany","product_name":"stand
ard","text_color":"#123456","text":"halooooo","picture_id":2386966,"font_id":24,"ship_
date":"2014-02-01"}]}
```

Rendered by yourself (as PDF)

If you want to customize all sides of your card, you can do so by sending us a pdf with all sides embedded in it. One pdf should contain one card. Here is an example of a valid standard postcard: prerendered.pdf

There are 2 additional mandatory parameters:

Parameter	Type	Default	Mandatory	Possible values
product_name	text		*	"standard","panorama","jumbo",...

country	text		*	"DE","US",... ISO2 Standard
pdf	file		*	-
ship_date	date			"2015-12-24"

The *country* is a necessary information for calculating the right postage for every card.

The dimensions for the PDF pages are listed here:

Format	Width	Height	Pages
standard	1819px (154mm or 6.062inch)	1311px (110mm or 4.331inch)	2
panorama	2551px (216mm or 8.5inch)	1311px (110mm or 4.331inch)	2
jumbo	2787px (236mm or 9.29inch)	1488px (126mm or 4.96inch)	2

These dimensions are optimal. If you can't match them exactly don't worry! We will zoom/shrink/crop the pages accordingly. However for best printquality you should stick with the required dimensions.

Example

```
order[cards_attributes][0][product_name]=standard
order[cards_attributes][0][country]=DE
order[cards_attributes][0][pdf]=@/prerendered-single-card.pdf
=> Status: 201, Body:
{"id":1950418,"cards":[{"id":1950777,"company":null,"title":null,"name":"Tester","street":null,"zip":null,"city":null,"state":null,"country":"Germany","product_name":"standard","text_color":"#123456","text":"halooooo","picture_id":2386966,"font_id":24,"ship_date":"2014-02-01"}]}
```

Testing

If your API key is in testing mode then all your orders are considered as tests automatically.

If your API key is in live mode then you are still able to send test-orders which won't be printed and charged by adding the parameter **test** with the value **1** to your request. This could be handy if you changed something in your api-call logic and want to test it.

Example

```
order[cards_attributes][0][product_name]=standard
order[cards_attributes][0][country]=DE
order[cards_attributes][0][pdf]=@/prerendered-single-card.pdf
test=1
```

Pictures

Own Pictues

All your transfered pictures can be reused for further orders if you use the "Rendered by Pokamax" method. This is handy in cases when you want to transfer multiple cards with the same picture and save bandwidth. So instead uploading the same picture over and over again, you can transfer it up-front and then reference it with the corresponding id.

Upload

Just send a POST Request to the **pictures** endpoint and you get as a response statuscode 201 and json with the id of the picture. The id won't change so it is a good practice to store it as a reference_id on your systems.

Example

```
picture[file]=@/picture.jpg
=> Status: 201, Body:
{"id":456,"thumb_url":"https://...","created_at":"2015-05-17T23:35:56.044+02:00"}
```

List PAGINATED

You can grab the list of all your uploaded pictures in your account by calling the **pictures** endpoint via GET:

```
=> Status: 201, Body:
[{"id":123,"thumb_url":"https://...","created_at":"2014-02-01T01:01:45.000+01:00"},...
]
```

You can get a thumbnail of a picture by calling the *thumb_url* via GET. Just don't forget the authorization parameter!

Delete

If you want to delete a picture from our servers then call the **pictures/{id}** endpoint via DELETE.

Gallery

PokaMax offers you access to our gallery with high quality designer pictures. However the access is limited to just the free pictures due to licence agreements with our designers. If you want access to all our gallery pictures then contact us. We can activate your api access to those pictures but we need to charge an extra fee for every ordered card with those pictures. You can distinguish them by analysing the *price* attribute. The price is defined in eurocent.

> ⓘ Keep in mind that pictures in our gallery could get deleted. If you reference to a deleted picture in an order you will receive an error.

List categories PAGINATED

Just call the **gallery/categories** endpoint via GET.

```
=> Status: 200, Body:
[{"id":100,"name":"Advent","locale":"de","picture_count":35},...]
```

List pictures PAGINATED

Just call the **gallery/categories/{category_id}/pictures** endpoint via GET.

```
=> Status: 200, Body:
[{"id":6662,"created_at":"2009-12-06T11:19:25.000+01:00","text":"let it snow, let it
snow, let it snow...!","locale":"de","thumb_url":"https://...","price":20},...]
```

Orders

List PAGINATED

Call the **orders** endpoint vie GET.

```
=> Status: 200, Body:
[{"id":1950418,"cards":[{"id":1950777,"company":null,"title":null,"name":"Tester","street":null,"zip":null,"city":null,"state":null,"country":"Germany","product_name":"standard","text_color":"#123456","text":"halooooo","picture_id":2386966,"font_id":24,"ship_date":"2014-02-01"}]}]
```

Pagination

All actions in this doc annotated with a **PAGINATED** -tag don't answer with all results. Think of the response as one single page of multiple pages. You can access them by appending the parameter *page* with a number to get the desired page.

Every response has also a header called X-Pagination with some useful pagination-data:

```
X-Pagination:
{"total":16,"total_pages":1,"first_page":true,"last_page":true,"current_page":1,"out_of_bounds":false,"offset":0}
```