



# DOCUMENTATION

Invoice Index

## Inhoud

1	API Documentation .....	2
2	Obtaining GUI account / API key .....	2
3	Using GUI account .....	2
4	Using API.....	2
4.1	cURL.....	2
4.2	PHP cURL .....	3
4.3	Java OK HTTP .....	3
4.4	Request response containing API token.....	4
5	Providing API token with later requests.....	4
5.1	cURL.....	4
5.2	PHP cURL .....	4
5.3	Java OK HTTP .....	5

## 1 API Documentation

Invoice Index's Application Programming Interface (API) is documented in Swagger. Swagger provides developers with a set of tools that help while developing an API. One of the tools they developed is a API documentation tool. The API documentation tool is easy to set-up for developers and makes implementing the API easier for the future API users. For example: Swagger (UI) documentation makes it possible for implementers to make and test requests on the fly through the Swagger UI. Apart from the possibility to make requests, Swagger UI visualizes the API in a standardized way that should be readable and understandable for everyone.

The Swagger UI can be found at the following URL:

<http://invoice-index.io/api/v1>

## 2 Obtaining GUI account / API key

In order to get access to the Invoice Index an GUI account and/or API key is required. Those can be requested by contacting Invoice Index on following email address:

[contact@invoice-index.io](mailto:contact@invoice-index.io)

After approval, requester will be supplied with a PIN code, API key and/or password.

## 3 Using GUI account

Invoice Index has a Graphical User Interface (GUI). In order to make use of the GUI a user has to login by submitting his username and password. After verifying the provided credentials access is or is not granted. The GUI is self-explanatory, some functions require the user to enter his PIN code.

The Invoice Index GUI can be found at the following URL:

<http://demo.invoice-index.io>

## 4 Using API

Using the API requires the user to authenticate himself with his username and API key, this way unauthorized parties are kept out. After authenticating, the user receives a API token that is valid for 900 seconds. This token has to be sent with every request the user makes.

Some authenticating coding examples are on the following page, along with the proper response.

### 4.1 cURL

```
curl -X POST \
http://invoice-index.io/api/v1/user/login \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d '{"cUsername" : "email@domain.com", "cApiKey" :
"04aVg8gnyjr3fs8vqsYcasd2a2fqvH0536c5gp9N4zVmyJc5"}'
```

## 4.2 PHP cURL

```
<?php
$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "http://invoice-index.io/api/v1/user/login",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "POST",
    CURLOPT_POSTFIELDS => "{\"cUsername\" : \"email@domain.com\", \"cApiKey\" : \"04aVg8gnyjr3fs8vqsYcasd2a2fqvH0536c5gp9N4zVmyJc5\"}",
    CURLOPT_HTTPHEADER => array(
        "Content-Type: application/json",
        "cache-control: no-cache"
    ),
));

$response = curl_exec($curl);
$error = curl_error($curl);

curl_close($curl);

if ($error) {
    echo "cURL Error #:" . $error;
} else {
    echo $response;
}
```

## 4.3 Java OK HTTP

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\"cUsername\" : \"email@domain.com\", \"cApiKey\" : \"04aVg8gnyjr3fs8vqsYcasd2a2fqvH0536c5gp9N4zVmyJc5\"}");
Request request = new Request.Builder()
    .url("http://invoice-index.io/api/v1/user/login")
    .post(body)
    .addHeader("Content-Type", "application/json")
    .addHeader("cache-control", "no-cache")
    .build();

Response response = client.newCall(request).execute();
```

## 4.4 Request response containing API token

```
{
  "aResourceResponse": {
    "bSuccess": true,
    "cMessage": "",
    "aToken": {
      "cToken":
"2720013cad206604f56ab9006da501de6acc334b5c762485c68483a1cfca07ee",
      "cTokenTimeValidSeconds": 900
    }
  }
}
```

## 5 Providing API token with later requests

The token is valid for 900 seconds, after 900 (as seen in above picture) seconds a new token has to be obtained by authenticating again. Except for the authenticate API call, all API requests require a API token to be sent with them in the header. The token needs to be set under the “Authorization” header.

Some requests with API token coding examples:

### 5.1 cURL

```
curl -X GET \
'http://invoice-index.io/api/v1/invoice?invoiceNr=431221&cocNr=4321' \
-H 'Authorization:
22ea9bb98090f53695700043a61f5004d05d7a353257e65464a7aad12ed9e028' \
-H 'cache-control: no-cache'
```

### 5.2 PHP cURL

```
<?php
$curl = curl_init();

curl_setopt_array($curl, array(
    CURLOPT_URL => "http://invoice-
index.io/api/v1/invoice?invoiceNr=431221&cocNr=4321",
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_ENCODING => "",
    CURLOPT_MAXREDIRS => 10,
    CURLOPT_TIMEOUT => 30,
    CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
    CURLOPT_CUSTOMREQUEST => "GET",
    CURLOPT_HTTPHEADER => array(
        "Authorization:
22ea9bb98090f53695700043a61f5004d05d7a353257e65464a7aad12ed9e028",
        "cache-control: no-cache"
    ),
));

$response = curl_exec($curl);
$error = curl_error($curl);

curl_close($curl);

if ($error) {
    echo "cURL Error #:" . $error;
} else {
    echo $response;
}
```

### 5.3 Java OK HTTP

```
OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
    .url("http://invoice-
index.io/api/v1/invoice?invoiceNr=431221&cocNr=4321")
    .get()
    .addHeader("Authorization",
"22ea9bb98090f53695700043a61f5004d05d7a353257e65464a7aad12ed9e028")
    .addHeader("cache-control", "no-cache")
    .build();

Response response = client.newCall(request).execute();
```