



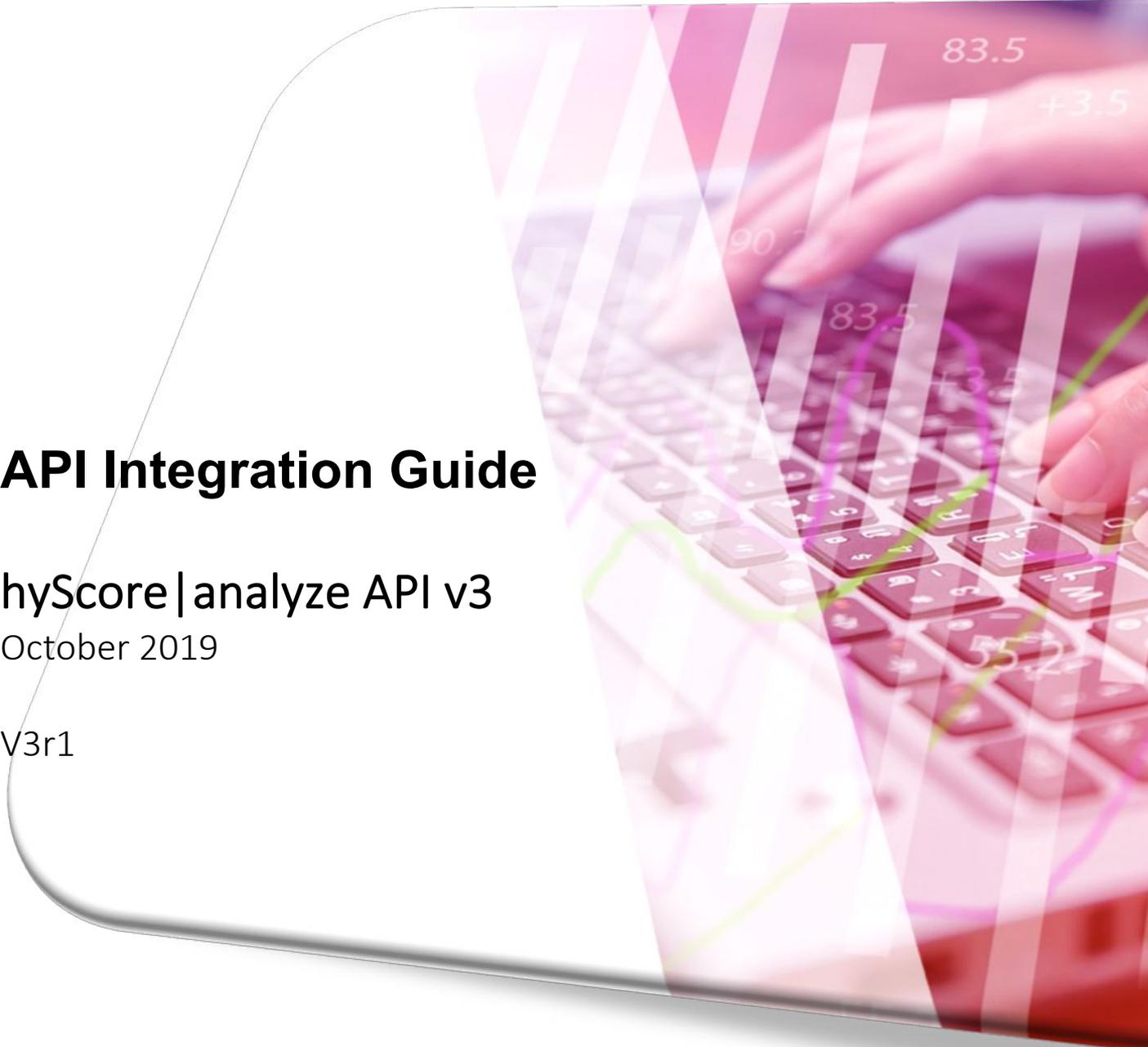
We understand content

API Integration Guide

hyScore | analyze API v3

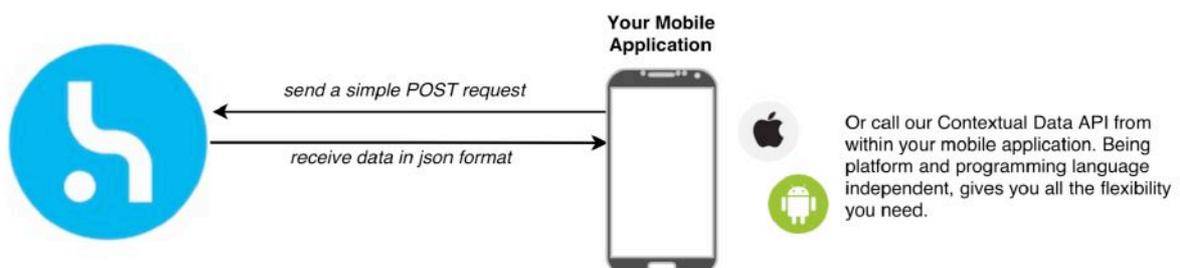
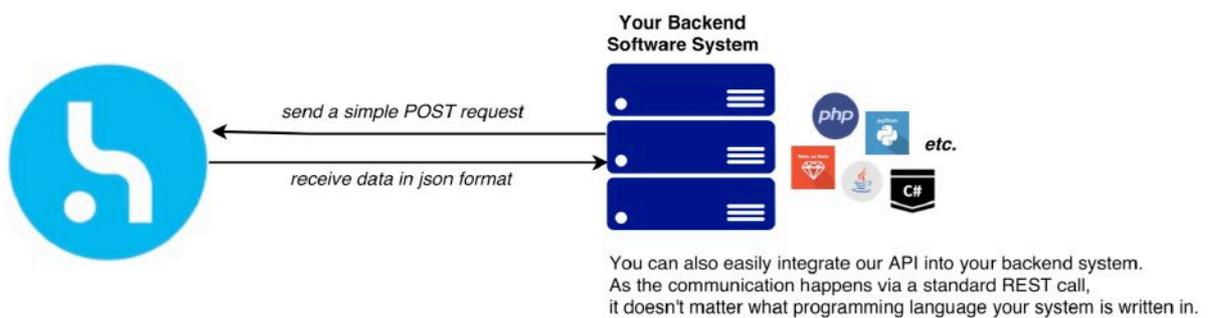
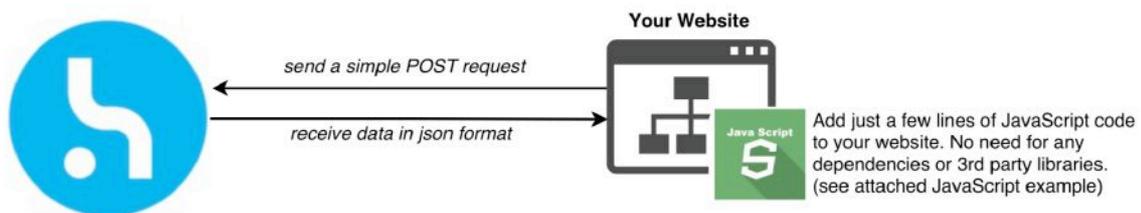
October 2019

V3r1



Overview

No Matter if you want to integrate our hyScore|analyze API into your website with JavaScript or call it from your backend systems written in Java, C#, Python, or any other language. Our Contextual Data API is platform and programming language independent and can be integrated in any system in just a few minutes. It's as simple as sending us a HTTP POST request and receiving the data in JSON format.



Use Cases

Real-Time vs Bulk

Most of our customers use cases can be divided into two different scenarios: real-time and bulk processing. For both the integration is exactly the same, what differs is how the data is “collected” and used.

Wanting to categorize and catalog your website inventory once, is an example for bulk processing. You might send us several thousand or even millions of URLs via the `api.hyscore.io/v3/url` endpoint.

In a one-time analysis scenario, we recommend splitting the API request in several packets.

Each URL need, if it is unknown to our system, for an initial analysis up to a second (incl. crawling) depending on amount of content extracted from the URL. As bigger amounts of URLs will take some time to process and analyze, you should split it up in several parts and wait e.g. 15 minutes to collect the results.

In a real-time scenario you send a URL and it gets analyzed as fast as can. In average the initial analysis takes a second. If you request the result faster, you might get an "analysis in progress" status. Once the result is available the result is cached and provided in milliseconds. The cached result will be updated with the latest information if the analysis methods change.

The nice thing is to collect the result is just sending us the same request again. This keeps the implementation effort for the integration very low.

By using the `api.hyscore.io/v3/text` endpoint the behavior is different. Here you send us plain text and the analysis is performed directly. That means that you get a result instantly. Depending on the text length an analysis can take several seconds.

If you've questions don't hesitate to contact our Support at [**support@hyscore.io**](mailto:support@hyscore.io)

How-To, Code Examples, and further Resources

Integrating our API is very simple and a matter of minutes. In the next pages we put together a quick-start tutorial and some code examples to make it as easy as possible for you. You can find further information, for example the full description of the response JSON, on our website at <https://hyscore.io/resources/>. If you have any questions or problems, just get in touch with us.

Full API documentation

You can find our full API specification here: [API Documentation](#)

Quick start

The Request

1. Create a POST request and add these headers:

```
x-api-key = YOUR_API_KEY  
Content-Type = application/json
```

2. Set the body to the URL that should be analyzed:

```
url = THE_URL_YOU_WANT_TO_HAVE_ANALYZED
```

Please note that the body information needs to be URL encoded, and must include the protocol, e.g. *http://* or *https://*

3. Send the POST request to our API endpoint:

```
https://api.hyscore.io/v3/url/
```

The Response

The response will be in JSON format. You can find the full description and examples of responses on our website: [API Documentation](#).

Successful

If we already know the requested URL/text or can analyze it in under 100ms, you will receive a successful response with all data.

No data yet

Depending on the size of a website, it's load time, and other factors, it might take us some time to fully analyze the contextual data. If we can't process your request in 100ms, we will send you a "No data yet" response. This way we make sure that your request is not blocked. Just send us the same request again after a short wait.

Categorization incomplete

Sometimes, it can take longer to determine the website category. In these cases, we send you a "Categorization issue" of type "incomplete". The response will already include all other data, but to get the category, you will need to send us the same request again after waiting some time.

Code Examples

JavaScript

```
function getData() {
  // configure the request
  const settings = {
    "async": true,
    "crossDomain": true,
    "url": "https://api.hyscore.io/v3/url",
    "method": "POST",
    "headers": {
      "x-api-key": YOUR_API_KEY,
      "content-type": "application/json"
    },
    "data":
      "url": http://www.example.com
      JSON.stringify({
    })
  }
  // send the request
  $.ajax(settings).done(function(response) {
    // extract the contextual data from 'response'
  });
}
```

Python

```
import requests
import json

url = "http://api.hyscore.io/v3/url"
headers = {
    'x-api-key': "YOUR_API_KEY",
    'Content-Type': "application/json"
}
payload = json.dumps({"url": "YOUR_URL"})
response = requests.request("POST", url, data=payload, headers=headers)
print(response.text)
```

Java

```
// create a POST request
URL url = new URL("https://api.hyscore.io/v3/url");
URLConnection con = (URLConnection) url.openConnection();
con.setRequestMethod("POST");

// set the header information
con.setRequestProperty("x-api-key", YOUR_API_KEY);
con.setRequestProperty("Content-Type", "application/json");

// set the body information
String body = "url=" + URLEncoder.encode("http://example.com", "UTF-8");

con.setDoOutput(true);
DataOutputStream out = new DataOutputStream(con.getOutputStream());
out.writeBytes(body);
out.flush();
out.close();

// send request
int status = con.getResponseCode();

// read response
InputStreamReader reader = new InputStreamReader(con.getInputStream());
    BufferedReader in = new BufferedReader(reader);
String inputLine;
StringBuffer content = new StringBuffer();
while ((inputLine = in.readLine()) != null) {
    content.append(inputLine);
}
in.close();
con.disconnect();

System.out.println(status); // the http status code of the request
System.out.println(content); // the contextual data as json
```