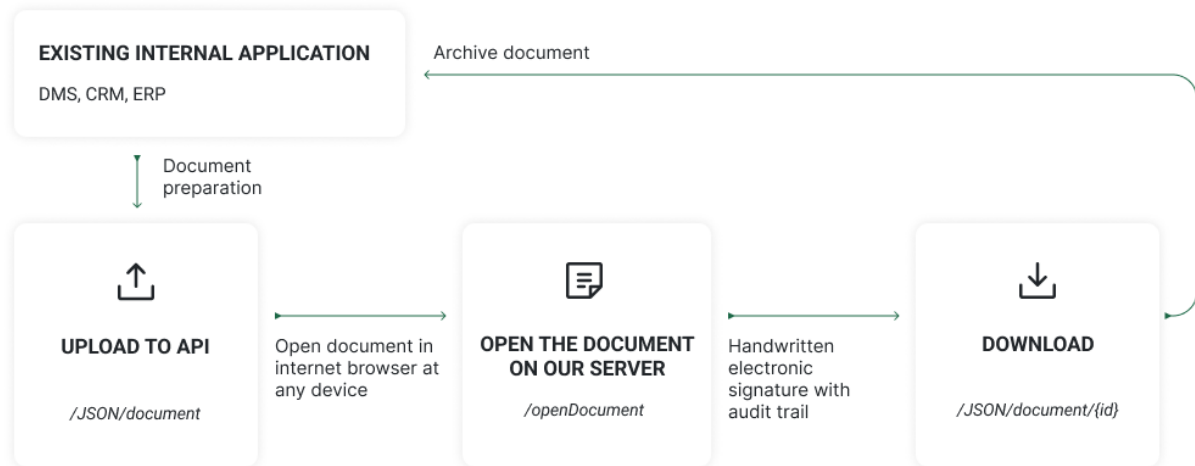


okdokument

JSON API BASIC INTEGRATION

INTEGRATION GUIDE



Introduction

API is a quick and simple way to make the best of our feature loaded online e-signature solution. A simple integration without your own web application – you are one step away from incorporating electronic signing into your digital environment.

We use JSON technology for the REST API to enable variability and ease of use across all the different options our service provides.

This document explains how the API works as well as the basic scenarios of how to sign a document, how to download a document and how to store it within internal systems.

Basic set of API calls consists of the following:

- 1) Upload document (JSON/Document)
- 2) Open document (/openDocument)
- 3) Download document (JSON/document/{id})

Before we start, there is one important parameter to explain – signatureRequestId.

SignatureRequestId is a unique identifier of document signing process, which has to be created before the signing process activation. It is possible to request signatureRequestId in a very simple way, just to register the document signing process, but it is also possible to define a lot of options for the document signing process during the signatureRequestId registration. Example below describes the simple way, more configuration options can be provided upon request.

1 CREATE SIGNATURE REQUEST ID

First step is to provide the document. Document is uploaded via “POST” call.

POST service JSON/signatureRequest is used to upload the PDF document and to define the signing process.

Service specifications:

URL	https://apidev.okdokument.com/signatureRequest
Method	POST
Content type	application/json
Response	HTTP 200 when uploaded successfully

JSON request body parts (Content type):

resulturl	URL to redirect the browser after finishing the signing process
signatureRequestId	Unique ID of the signatureRequest, that is used for accessing document for sign via /openDocument service
content	The binary content of the uploaded PDF file in base64 format

Request body example

```
"filedata":{"content":"xyz_JVBERi0xLjc=="}}
```

Response body example

```
{
  "message": "msg.signature.request.id.created",
  "result": "y7Ki_u1mrUwijfb-o_S1JiphSsoXya8iypH380F",
  "url": "https://apidev.okdokument.com/openDocument/9126dbbd-d2bb-5f78-a4b6-e291396",
}
```

2 OPEN DOCUMENT

Next step is to open the document for signature. Document is opened by calling the “openDocument” service. Service displays the PDF document in our GUI for a signature.

After the user signs the document it can be sent back or to another destination where it should be stored. Once done, user is redirected to a defined URL (e.g. back to your application).

Service specifications:

URL	https://apidev.okdokument.com/openDocument/{signatureRequestId}
Method	GET
URL parameter signatureRequestId	unique identifier used in /JSON/document method
Response	HTTP status 302 when PDF document is uploaded and processed successfully, user is transferred to the document view HTTP status 400 when signatureRequestId is invalid

Example:

<https://apidev.okdokument.com/openDocument/9126dbbd-d2bb-5f78-a4b6-e291396>

Response header example:

HTTP/1.1 302
Location: https://apidev.okdokument.com/viewDocument?id=y7Ki_u1mrUwijfb-o_S1JiphSsoXya8iypH380F

3 DOWNLOAD A DOCUMENT

Once the document is signed it may be downloaded. GET method /JSON/document/{id} is used to download the (binary) content of a signed document.

Service specifications:

URL	https://apidev.okdokument.com/JSON/document/{id}
Method	GET
Response	HTTP 200 when successful

Example:

https://apidev.okdokument.com/JSON/document/y7Ki_u1mrUwijfb-o_S1JiphSsoXya8iypH380F