

How to convert PDF to text from file (node for PDF to text API in JavaScript and PDF.co Web API)

How to convert PDF to text from file (node in JavaScript with easy ByteScout code samples to make PDF to text API. Step-by-step tutorial

The documentation is designed to help you to implement the features on your side. PDF.co Web API was made to help with PDF to text API in JavaScript. PDF.co Web API is the Rest API that provides set of data extraction functions, tools for documents manipulation, splitting and merging of pdf files. Includes built-in OCR, images recognition, can generate and read barcodes from images, scans and pdf.

JavaScript code samples for JavaScript developers help to speed up the application's code writing when using PDF.co Web API. This JavaScript sample code should be copied and pasted into your project. After doing this just compile your project and click Run. Enjoy writing a code with ready-to-use sample JavaScript codes to implement PDF to text API using PDF.co Web API.

Trial version of ByteScout is available for free download from our website. This and other source code samples for JavaScript and other programming languages are available.

JavaScript - ConvertPdfToTextFromFile_AsyncApi.js

```
/*jshint esversion: 6 */

var https = require("https");
var path = require("path");
var fs = require("fs");

// `request` module is required for file upload.
// Use "npm install request" command to install.
var request = require("request");

// The authentication key (API Key).
// Get your own by registering at https://app.pdf.co/documentation/api
const API_KEY = "*****";

// Source PDF file
const SourceFile = "./sample.pdf";
// Comma-separated list of page indices (or ranges) to process. Leave empty for all
pages. Example: '0,2-5,7-'.
const Pages = "";
// PDF document password. Leave empty for unprotected documents.
const Password = "";
// Destination TXT file name
const DestinationFile = "./result.txt";
```

```

// Prepare request uri for `PDF To Text` API endpoint
var query = `https://api.pdf.co/v1/pdf/convert/to/text`;
let reqOptions = {
  uri: query,
  headers: { "x-api-key": API_KEY },
  formData: {
    name: path.basename(DestinationFile),
    pages: Pages,
    password: Password,
    async: 'True',
    file: fs.createReadStream(SourceFile)
  },
};

// Send request
request.post(reqOptions, function (error, response, body) {
  if (error) {
    return console.error("Error: ", error);
  }

  // Parse JSON response
  let data = JSON.parse(body);
  console.log(`Job #${data.jobId} has been created!`);
  checkIfJobIsCompleted(data.jobId, data.url);
});

function checkIfJobIsCompleted(jobId, resultFileUrl) {
  let queryPath = `/v1/job/check?jobid=${jobId}`;
  let reqOptions = {
    host: "api.pdf.co",
    path: encodeURI(queryPath),
    method: "GET",
    headers: { "x-api-key": API_KEY }
  };

  https.get(reqOptions, (response) => {
    response.on("data", (d) => {
      response.setEncoding("utf8");

      // Parse JSON response
      let data = JSON.parse(d);
      console.log(`Checking Job #${jobId}, Status: ${data.status}, Time: ${new
Date().toLocaleString()}`);

      if (data.status == "working") {
        // Check again after 3 seconds
        setTimeout(function(){
          checkIfJobIsCompleted(jobId, resultFileUrl);
        }, 3000);
      }
      else if (data.status == "success") {
        // Download TXT file
        var file = fs.createWriteStream(DestinationFile);
        https.get(resultFileUrl, (response2) => {
          response2.pipe(file)
            .on("close", () => {
              console.log(`Generated TXT file saved as
"${DestinationFile}" file.`);
            });
        });
      }
    });
  });
}

```

```
        });  
    }  
    else {  
        console.log(`Operation ended with status: "${data.status}".`);  
    }  
    })  
});  
}
```

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about PDF.co Web API](#)

[Explore documentation](#)

[Visit www.ByteScout.com](#)

or

[Get Your Free API Key for www.PDF.co Web API](#)