

How to convert PDF to XLSX from uploaded file (node for PDF to excel API in JavaScript with PDF.co Web API)

Tutorial: how to convert PDF to XLSX from uploaded file (node for PDF to excel API in JavaScript)

Sample source codes below will show you how to cope with a difficult task, for example, PDF to excel API in JavaScript. PDF.co Web API was made to help with PDF to excel API in JavaScript. PDF.co Web API is the Web API with a set of tools for documents manipulation, data conversion, data extraction, splitting and merging of documents. Includes image recognition, built-in OCR, barcode generation and barcode decoders to decode bar codes from scans, pictures and pdf.

JavaScript code samples for JavaScript developers help to speed up the application's code writing when using PDF.co Web API. Follow the instruction and copy - paste code for JavaScript into your project's code editor. This basic programming language sample code for JavaScript will do the whole work for you in implementing PDF to excel API in your app.

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 - ConvertPdfToXlsxFromUploadedFile.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 XLSX file name
```

```

const DestinationFile = "./result.xlsx";

// 1. RETRIEVE PRESIGNED URL TO UPLOAD FILE.
getPresignedUrl(API_KEY, SourceFile)
.then(([uploadUrl, uploadedFileUrl]) => {
  // 2. UPLOAD THE FILE TO CLOUD.
  uploadFile(API_KEY, SourceFile, uploadUrl)
  .then(() => {
    // 3. CONVERT UPLOADED PDF FILE TO XLSX
    convertPdfToXlsx(API_KEY, uploadedFileUrl, Password, Pages, DestinationFile);
  })
  .catch(e => {
    console.log(e);
  });
});

})
.catch(e => {
  console.log(e);
});

function getPresignedUrl(apiKey, localFile) {
  return new Promise(resolve => {
    // Prepare request to `Get Presigned URL` API endpoint
    let queryPath = `/v1/file/upload/get-presigned-url?
contentType=application/octet-stream&name=${path.basename(SourceFile)}`;
    let reqOptions = {
      host: "api.pdf.co",
      path: encodeURI(queryPath),
      headers: { "x-api-key": API_KEY }
    };
    // Send request
    https.get(reqOptions, (response) => {
      response.on("data", (d) => {
        let data = JSON.parse(d);
        if (data.error == false) {
          // Return presigned url we received
          resolve([data.presignedUrl, data.url]);
        }
        else {
          // Service reported error
          console.log("getPresignedUrl(): " + data.message);
        }
      });
    });
  })
  .on("error", (e) => {
    // Request error
    console.log("getPresignedUrl(): " + e);
  });
});

}

function uploadFile(apiKey, localFile, uploadUrl) {
  return new Promise(resolve => {
    fs.readFile(SourceFile, (err, data) => {
      request({
        method: "PUT",
        url: uploadUrl,
        body: data,
      })
    });
  });
}

```

```

        headers: {
            "Content-Type": "application/octet-stream"
        }
    }, (err, res, body) => {
        if (!err) {
            resolve();
        }
        else {
            console.log("uploadFile() request error: " + e);
        }
    });
});
});
}

```

```

function convertPdfToXlsx(apiKey, uploadedFileUrl, password, pages, destinationFile)
{
    // Prepare request to `PDF To XLSX` API endpoint
    var queryPath = `/v1/pdf/convert/to/xlsx?
name=${path.basename(destinationFile)}&password=${password}&pages=${pages}&url=${upload

    let reqOptions = {
        host: "api.pdf.co",
        path: encodeURI(queryPath),
        method: "GET",
        headers: { "x-api-key": API_KEY }
    };
    // Send request
    https.get(reqOptions, (response) => {
        response.on("data", (d) => {
            response.setEncoding("utf8");

            // Parse JSON response
            let data = JSON.parse(d);

            if (data.error == false) {
                console.log(`Job #${data.jobId} has been created!`);
                checkIfJobIsCompleted(data.jobId, data.url, destinationFile);
            }
            else {
                // Service reported error
                console.log("convertPdfToXlsx(): " + data.message);
            }
        });
    });
    .on("error", (e) => {
        // Request error
        console.log("convertPdfToXlsx(): " + e);
    });
}

```

```

function checkIfJobIsCompleted(jobId, resultFileUrl, destinationFile) {
    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, destinationFile);}, 3000);
    }
    else if (data.status == "success") {
      // Download XLSX file
      var file = fs.createWriteStream(destinationFile);
      https.get(resultFileUrl, (response2) => {
        response2.pipe(file)
          .on("close", () => {
            console.log(`Generated XLSX 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](#)