

www.bytescout.com

How to split PDF by text from uploaded file (node for PDF splitting API in JavaScript and PDF.co Web API

PDF.co Web API is the flexible Web API that includes full set of functions from e-signature requests to data extraction, OCR, images recognition, pdf splitting and pdf splitting. Can also generate barcodes and read barcodes from images, scans and pdf.

FOR MORE INFORMATION AND FREE TRIAL:

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

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

[Explore API Documentation](#)

[Get Free Training for PDF.co Web API](#)

[Get Free API key for Web API](#)

[visit www.ByteScout.com](http://www.ByteScout.com)

Source Code Files:

app.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 to split
const SourceFile = "./sample.pdf";
// Split Search String
const SplitText = "invoice number";

// 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. SPLIT UPLOADED PDF
        splitPdf(API_KEY, uploadedFileUrl, SplitText);
      })
      .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`;
    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: " + err);
        }
      });
    });
  });
}

function splitPdf(apiKey, uploadedFileUrl, splitText) {
  // Prepare request to `Split PDF By Text` API endpoint
  var queryPath = `/v1/pdf/split2`;

  // JSON payload for api request
  var jsonPayload = JSON.stringify({
    searchString: splitText, url: uploadedFileUrl, async: true
  });

  var reqOptions = {
    host: "api.pdf.co",
    method: "POST",
    path: queryPath,
    headers: {
      "x-api-key": apiKey,
      "Content-Type": "application/json",
      "Content-Length": Buffer.byteLength(jsonPayload, 'utf8')
    }
  };

  // Send request
  var postRequest = https.request(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);
      }
    }
  });
}

```

```

    else {
      // Service reported error
      console.log("splitPdf(): " + data.message);
    }
  });
}

.on("error", (e) => {
  // Request error
  console.log("splitPdf(): " + e);
});

// Write request data
postRequest.write(jsonPayload);
postRequest.end();
}

function checkIfJobIsCompleted(jobId, resultFileUrlJson) {
  let queryPath = `/v1/job/check`;

  // JSON payload for api request
  let jsonPayload = JSON.stringify({
    jobId: jobId
  });

  let reqOptions = {
    host: "api.pdf.co",
    path: queryPath,
    method: "POST",
    headers: {
      "x-api-key": API_KEY,
      "Content-Type": "application/json",
      "Content-Length": Buffer.byteLength(jsonPayload, 'utf8')
    }
  };

  // Send request
  var postRequest = https.request(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()}`);

      if (data.status == "working") {
        // Check again after 3 seconds
        setTimeout(function () { checkIfJobIsCompleted(jobId, resultFileUrlJson); }, 3000);
      }
      else if (data.status == "success") {

        request({ method: 'GET', uri: resultFileUrlJson, gzip: true },
          function (error, response, body) {

            // Parse JSON response
            let respJsonFileArray = JSON.parse(body);
            let part = 1;

            respJsonFileArray.forEach((url) => {
              var localFileName = `./part${part}.pdf`;
              var file = fs.createWriteStream(localFileName);
              https.get(url, (response2) => {

```

```

        response2.pipe(file)
        .on("close", () => {
            console.log(`Generated PDF file saved as "${lo
        });
    });
    part++;
}, this);

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

// Write request data
postRequest.write(jsonPayload);
postRequest.end();
}

```

package.json

```

{
  "name": "test",
  "version": "1.0.0",
  "description": "PDF.co",
  "main": "app.js",
  "scripts": {
  },
  "keywords": [
    "pdf.co",
    "web",
    "api",
    "bytescout",
    "api"
  ],
  "author": "ByteScout & PDF.co",
  "license": "ISC",
  "dependencies": {
    "request": "^2.88.2"
  }
}

```

VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit PDF.co Web API Home Page](#)

[Explore PDF.co Web API Documentation](#)

[Explore Samples](#)

[Sign Up for PDF.co Web API Online Training](#)

ON-DEMAND REST WEB API

[Get Your API Key](#)

[Explore Web API Docs](#)

[Explore Web API Samples](#)

[visit www.ByteScout.com](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com