

www.bytescout.com

delete text from PDF in JavaScript and ByteScout Cloud API Server

ByteScout Cloud API Server is the ready to deploy Web API Server that can be deployed in less than thirty minutes into your own in-house Windows server (no Internet connection is required to process data!) or into private cloud server. Can store data on in-house local server based storage or in Amazon AWS S3 bucket. Processing data solely on the server using built-in ByteScout powered engine, no cloud services are used to process your data!.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about **ByteScout Cloud API Server**](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Cloud API Server](#)

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

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

Source Code Files:

app.js

```

// Please NOTE: In this sample we're assuming Cloud Api Server is hosted at "https://l
// If it's not then please replace this with with your hosting url.

/*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";
// PDF document password. Leave empty for unprotected documents.
const Password = "";
// Destination PDF file name
const DestinationFile = "./result.pdf";

// 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. DELETE TEXT FROM UPLOADED PDF FILE
        deleteTextFromPdf(API_KEY, uploadedFileUrl, Password, 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
    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 deleteTextFromPdf(apiKey, uploadedFileUrl, password, destinationFile) {
    // Prepare request to `Delete Text from PDF` API endpoint
    var queryPath = `/v1/pdf/edit/delete-text`;
    // JSON payload for api request
    var jsonPayload = JSON.stringify({
        name: path.basename(destinationFile), password: password, url: uploadedFileUrl
    });

    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);
        console.log(`Job #${data.jobId} has been created!`);

        if (data.error == false) {
            checkIfJobIsCompleted(data.jobId, data.url, destinationFile);
        }
        else {
            // Service reported error
            console.log("readBarcodes(): " + data.message);
        }
    });
}

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

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

function checkIfJobIsCompleted(jobId, resultFileUrl, destinationFile) {
    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().toLocaleTimeString()}`);

            if (data.status == "working") {
                // Check again after 3 seconds
                setTimeout(function () { checkIfJobIsCompleted(jobId, resultFileUrl, destinationFile); }, 3000);
            }
            else if (data.status == "success") {
                // Download PDF file
                var file = fs.createWriteStream(destinationFile);
                https.get(resultFileUrl, (response2) => {
                    response2.pipe(file)
                })
            }
        });
    });
}

```

```

        .on("close", () => {
            console.log(`Generated PDF file saved as "${destinationFile}"`);
        });
    }
    else {
        console.log(`Operation ended with status: "${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 ByteScout Cloud API Server Home Page](#)
[Explore ByteScout Cloud API Server Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout Cloud API Server 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