www.bytescout.com

How to convert PDF to XML from uploaded file (node for PDF to XML API in JavaScript and ByteScout Cloud API Server

Step By Step Instructions on how to convert PDF to XML from uploaded file (node for PDF to XML API in JavaScript

This page displays the code samples for programming in JavaScript. PDF to XML API in JavaScript can be applied with ByteScout Cloud API Server. ByteScout Cloud API Server is API server that is ready to use and can be installed and deployed in less than 30 minutes on your own Windows server or server in a cloud. It can save data and files on your local server-based file storage or in Amazon AWS S3 storage. Data is processed solely on the API server and is powered by ByteScout engine, no cloud services or Internet connection is required for data processing..

Use the code displayed below in your application to save a lot of time on writing and testing code. Open your JavaScript project and simply copy & paste the code and then run your app! Check JavaScript sample code examples to see if they respond to your needs and requirements for the project.

Our website provides free trial version of ByteScout Cloud API Server that gives source code samples to assist with your JavaScript project.

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

Source Code Files:

ConvertPdfToXmlFromUploadedFile.js

```
/*jshint esversion: 6 */

// 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.

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");

// Source PDF file
const SourceFile = "./sample.pdf";
// Comma-separated list of page indices (or ranges) to process. Leave empty for all pa
const Pages = "";
// PDF document password. Leave empty for unprotected documents.
const Password = "";
// Destination XML file name
const DestinationFile = "./result.xml";


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


function getPresignedUrl(localFile) {
    return new Promise(resolve => {
        // Prepare request to `Get Presigned URL` API endpoint
        let queryPath = `/file/upload/get-presigned-url?contenttype=application/octet-
        let reqOptions = {
            host: "localhost",
            path: encodeURI(queryPath)
        };
        // 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(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 convertPdfToXml(uploadedFileUrl, password, pages, destinationFile) {
    // Prepare request to `PDF To XML` API endpoint
    var queryPath = `/pdf/convert/to/xml?name=${path.basename(destinationFile)}&passwo
    let reqOptions = {
        host: "localhost",
        path: encodeURI(queryPath),
        method: "GET"
    };
    // Send request
    https.get(reqOptions, (response) => {
        response.on("data", (d) => {
            response.setEncoding("utf8");
            // Parse JSON response
            let data = JSON.parse(d);
            if (data.error == false) {
                // Process returned job
                console.log(`Job #${data.jobId} has been created!`);
                checkIfJobIsCompleted(data.jobId, data.url, destinationFile);
            }
            else {
                // Service reported error
                console.log("convertPdfToXml(): " + data.message);
            }
```

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

function checkIfJobIsCompleted(jobId, resultFileUrl, destinationFile) {
    let queryPath = `/job/check?jobid=${jobId}`;
    let reqOptions = {
        host: "localhost",
        path: encodeURI(queryPath),
        method: "GET"
    };

    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 Da

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

VIDEO

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