

www.bytescout.com

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

Follow this simple tutorial to learn convert PDF to HTML from uploaded file (node to have PDF to HTML API in JavaScript

This page displays the step-by-step instructions and algorithm of how to convert PDF to HTML from uploaded file (node and how to apply it in your application. ByteScout Cloud API Server was designed to assist PDF to HTML API in JavaScript. ByteScout Cloud API Server is the ready to use Web API Server that can be deployed in less than 30 minutes into your own in-house server 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!.

If you want to speed up the application's code writing then JavaScript code samples for JavaScript developers help to implement using ByteScout Cloud API Server. This JavaScript sample code can be used by copying and pasting into your project. Once done, just compile your project and click Run. Check JavaScript sample code examples to see if they respond to your needs and requirements for the project.

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.

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:

ConvertPdfToHtmlFromUploadedFile.js

```
/*jshint esversion: 6 */

// Please NOTE: In this sample we're assuming Cloud Api Server is hosted at "https://localhost:9443"
// If it's not then please replace this 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 pages.
const Pages = "";
// PDF document password. Leave empty for unprotected documents.
const Password = "";
// Destination HTML file name
const DestinationFile = "./result.html";
// Set to `true` to get simplified HTML without CSS. Default is the rich HTML keeping all styles.
const PlainHtml = false;
// Set to `true` if your document has the column layout like a newspaper.
const ColumnLayout = false;

// 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 HTML
        convertPdfToHtml(uploadedFileUrl, Password, Pages, PlainHtml, ColumnLayout, 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-stream`;
        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 convertPdfToHtml(uploadedFileUrl, password, pages, plainHtml, columnLayout, de
// Prepare request to `PDF To HTML` API endpoint
var queryPath = `/pdf/convert/to/html?name=${path.basename(destinationFile)}&passwo
`&simple=${plainHtml}&columns=${columnLayout}&url=${uploadedFileUrl}&async=True
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);
        console.log(`Job #${data.jobId} has been created!`);
    });
});

```

```

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

function checkIfJobIsCompleted(jobId, resultImageUrl, 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 Date().toLocaleString()}`);

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

```

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](#)

[visit www.PDF.co](#)

[www.bytescout.com](#)