

How to optimize PDF from uploaded file (node for PDF optimization API in JavaScript using ByteScout Cloud API Server)

Learn in simple ways: How to optimize PDF from uploaded file (node for PDF optimization API in JavaScript

This page displays the code samples for programming in JavaScript. PDF optimization API in JavaScript can be applied with 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!.

JavaScript code snippet like this for ByteScout Cloud API Server works best when you need to quickly implement PDF optimization API in your JavaScript application. Open your JavaScript project and simply copy & paste the code and then run your app! Writing JavaScript application mostly includes various stages of the software development so even if the functionality works please check it with your data and the production environment.

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:

OptimizePdfFromUploadedFile.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";
// 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(SourceFile)
.then(([uploadUrl, uploadedFileUrl]) => {
    // 2. UPLOAD THE FILE TO CLOUD.
    uploadFile(SourceFile, uploadUrl)
    .then(() => {
        // 3. OPTIMIZE UPLOADED PDF FILE
        optimizePdf(uploadedFileUrl, Password, 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.url);
                }
            });
        });
    });
}
```

```

        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 optimizePdf(uploadedFileUrl, password, destinationFile) {
    // Prepare request to `Optimize PDF` API endpoint
    var queryPath = `/pdf/optimize?name=${path.basename(destinationFile)}&password=${password}`;
    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("readBarcodes(): " + data.message);
            }
        });
    });
}

```

```

        });
    });
    .on("error", (e) => {
        // Request error
        console.log(`readBarcodes(): ${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 PDF file
                var file = fs.createWriteStream(destinationFile);
                https.get(resultImageUrl, (response2) => {
                    response2.pipe(file)
                        .on("close", () => {
                            console.log(`Generated PDF 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](#)