

www.bytescout.com

How to merge PDF documents from uploaded file (node for PDF merging API in JavaScript with ByteScout Cloud API Server)

Learn in simple ways: How to merge PDF documents from uploaded file (node for PDF merging API in JavaScript

Every ByteScout tool includes sample JavaScript source codes that you can find here or in the folder with installed ByteScout product. ByteScout Cloud API Server was designed to assist PDF merging 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!.

This simple and easy to understand sample source code in JavaScript for ByteScout Cloud API Server contains different functions and options you should do calling the API to implement PDF merging API. This sample code in JavaScript is all you need. Just copy-paste it to the code editor, then add a reference to ByteScout Cloud API Server and you are ready to try it! Use of ByteScout Cloud API Server in JavaScript is also described in the documentation given along with the product.

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:

MergePDFDocumentsFromUploadedFile.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 SourceFile1 = "./sample1.pdf";
const SourceFile2 = "./sample2.pdf";

// Destination PDF file name
const DestinationFile = "./result.pdf";

// Upload File-1: - 1. RETRIEVE PRESIGNED URL TO UPLOAD FILE.
getPresignedUrl(SourceFile1)
    .then(([uploadUrl1, uploadedFileUrl1]) => {

        // Upload File-1: - 2. UPLOAD THE FILE TO CLOUD.
        uploadFile(SourceFile1, uploadUrl1)
            .then(() => {

                // Upload File-2: - 1. RETRIEVE PRESIGNED URL TO UPLOAD FILE.
                getPresignedUrl(SourceFile2)
                    .then(([uploadUrl2, uploadedFileUrl2]) => {

                        // Upload File-2: - 2. UPLOAD THE FILE TO CLOUD.
                        uploadFile(SourceFile2, uploadUrl2)
                            .then(() => {

                                const SourceFiles = [
                                    uploadedFileUrl1,
                                    uploadedFileUrl2
                                ];

                                // Perform Merge PDF Documents
                                mergePDFDocuments(SourceFiles, DestinationFile);
                            })
                            .catch(e => {
                                console.log(e);
                            });
                    })
                    .catch(e => {
                        console.log(e);
                    });
            });
    })
    .catch(e => {
        console.log(e);
    });
})
```

```

        .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(sourceFile, 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 mergePDFDocuments(SourceFiles, DestinationFile) {
    // Prepare request to `Merge PDF` API endpoint
    var queryPath = `/pdf/merge?name=${path.basename(DestinationFile)}&url=${SourceFile}`;
    var reqOptions = {
        host: "localhost",
        path: encodeURI(queryPath)
    };
    // Send request
    https.get(reqOptions, (response) => {
        response.on("data", (d) => {
            // Parse JSON response
            var 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(data.message);
            }
        });
    }).on("error", (e) => {
        // Request error
        console.log(e);
    });
}

function checkIfJobIsCompleted(jobId, resultImageUrl) {
    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); }, 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](#)