

## How to merge PDF documents from uploaded file (node for PDF merging API in JavaScript with PDF.co Web API)

Learn how to merge PDF documents from uploaded file (node to have PDF merging API in JavaScript)

The coding tutorials are designed to help you test the features without need to write your own code. PDF merging API in JavaScript can be implemented with PDF.co Web API. PDF.co Web API is the Web API with a set of tools for documents manipulation, data conversion, data extraction, splitting and merging of documents. Includes image recognition, built-in OCR, barcode generation and barcode decoders to decode bar codes from scans, pictures and pdf.

JavaScript code samples for JavaScript developers help to speed up the application's code writing when using PDF.co Web API. Follow the instruction and copy - paste code for JavaScript into your project's code editor. Test JavaScript sample code examples whether they respond your needs and requirements for the project.

PDF.co Web API - free trial version is on available our website. Also, there are other code samples to help you with your JavaScript application included into trial version.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about PDF.co Web API](#)

[Explore API Documentation](#)

[Get Free Training for PDF.co Web API](#)

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

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

Source Code Files:

## MergePDFDocumentsFromUploadedFile.js

```
/*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 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);
      });
    });
```

```

    });
  })
  .catch(e => {
    console.log(e);
  });

function getPresignedUrl(localFile) {
  return new Promise(resolve => {
    // Prepare request to `Get Presigned URL` API endpoint
    let queryPath = `/v1/file/upload/get-presigned-url?contenttype=application/octet-stream`;
    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(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 = `/v1/pdf/merge?name=${path.basename(DestinationFile)}&url=${SourceFile}`;
var reqOptions = {
  host: "api.pdf.co",
  path: encodeURI(queryPath),
  headers: {
    "x-api-key": API_KEY
  }
};
// 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, resultFileUrl) {
  let queryPath = `/v1/job/check?jobid=${jobId}`;
  let reqOptions = {
    host: "api.pdf.co",
    path: encodeURI(queryPath),
    method: "GET",
    headers: { "x-api-key": API_KEY }
  };

  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().toLocaleTimeString()}`);

      if (data.status == "working") {
        // Check again after 3 seconds
        setTimeout(function() { checkIfJobIsCompleted(jobId, resultFileUrl); }, 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}".`);  
        }  
    });  
}
```

---

## VIDEO

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

## ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit PDF.co Web API Home Page](#)  
[Explore PDF.co Web API Documentation](#)  
[Explore Samples](#)  
[Sign Up for PDF.co Web API 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](http://www.bytescout.com)