

[www.bytescout.com](http://www.bytescout.com)

## How to parse multipage table for document parser API in PHP using 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.

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:

MultiPageTable-template1.yml

```
---
# Template that demonstrates parsing of multi-page table using only
# macro expressions for the table start, end, and rows.
# If macro expression cannot be written for every table row (for example,
# if the table contains empty cells), try the second method demonstrated
# in `MultiPageTable-template2.yml` template.
templateVersion: 3
templatePriority: 0
sourceId: Multipage Table Test
detectionRules:
  keywords:
    - Sample document with multi-page table
fields:
  total:
    expression: 'TOTAL_{{Spaces}}_{{Number}}'
    dataType: decimal
```

```

tables:
- name: table1
  start:
    # macro expression to find the table start in document
    expression: 'Item{{Spaces}}Description{{Spaces}}Price'
  end:
    # macro expression to find the table end in document
    expression: 'TOTAL{{Spaces}}{Number}'
  row:
    # macro expression to find table rows
    expression: '{{LineStart}}{Spaces}{?<itemNo>{Digits}}{Spaces}{?<description>{SentenceWithSingleSpaces}}{
# output data types for columns
columns:
- name: itemNo
  type: integer
- name: description
  type: string
- name: price
  type: decimal
- name: qty
  type: integer
- name: extPrice
  type: decimal
multipage: true

```

## MultiPageTable-template2.yml

```

---
# Template that demonstrates parsing of multi-page table without using
# macro expression for table rows. Rows and cells are extracted automatically
# by specified column coordinates. Use `Template Editor` app to find the coordinates
# (coordinates of the mouse cursor are displayed in the toolbar).
templateVersion: 3
templatePriority: 0
sourceld: Multipage Table Test
detectionRules:
  keywords:
  - Sample document with multi-page table
fields:
total:
  type: regex
  expression: 'TOTAL{{Spaces}}{Number}'
  dataType: decimal
tables:
- name: table1
  # coordinate OR macro expression to find the table start on each document page
  start:
    #y: 136
    expression: 'Item{{Spaces}}Description{{Spaces}}Price'
  # coordinate OR macro expression to find the table end on each document page
  end:
    #y: 787
    expression: '(Page {Digits} of {Digits})(TOTAL{{Spaces}}{Number})'
  # left coordinate of the table (optional)
  left: 51
  # right coordinate of the table (optional)
  right: 528
  # column names, output data types and coordinates (left column edge)
  columns:
  - name: itemNo
    x: 51
    type: integer
  - name: description
    x: 102
    type: string
  - name: price
    x: 324

```

```
type: decimal
- name: qty
  x: 396
type: integer
- name: extPrice
  x: 441
type: decimal
multipage: true
```

program.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document Parse Results</title>
</head>
<body>

<?php

// Get submitted form data
$apiKey = $_POST["apiKey"]; // The authentication key (API Key). Get your own by registering at https://app.pdf.co/d

// 1. RETRIEVE THE PRESIGNED URL TO UPLOAD THE FILE.
// * If you already have the direct PDF file link, go to the step 3.

// Create URL
$url = "https://api.pdf.co/v1/file/upload/get-presigned-url" .
      "?name=" . $_FILES["file"]["tmp_name"] .
      "&contentType=application/octet-stream";

// Create request
$curl = curl_init();
curl_setopt($curl, CURLOPT_HTTPHEADER, array("x-api-key: " . $apiKey));
curl_setopt($curl, CURLOPT_URL, $url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
// Execute request
$result = curl_exec($curl);

if (curl_errno($curl) == 0)
{
  $status_code = curl_getinfo($curl, CURLINFO_HTTP_CODE);

  if ($status_code == 200)
  {
    $json = json_decode($result, true);

    // Get URL to use for the file upload
    $uploadFileUrl = $json["presignedUrl"];
    // Get URL of uploaded file to use with later API calls
    $uploadedFileUrl = $json["url"];

    // 2. UPLOAD THE FILE TO CLOUD.

    $localFile = $_FILES["fileInput"]["tmp_name"];
    $fileHandle = fopen($localFile, "r");

    curl_setopt($curl, CURLOPT_URL, $uploadFileUrl);
    curl_setopt($curl, CURLOPT_HTTPHEADER, array("content-type: application/octet-stream"));
    curl_setopt($curl, CURLOPT_PUT, true);
    curl_setopt($curl, CURLOPT_INFILE, $fileHandle);
    curl_setopt($curl, CURLOPT_INFILESIZE, filesize($localFile));

    // Execute request
```

```

curl_exec($curl);

fclose($fileHandle);

if (curl_errno($curl) == 0)
{
    $status_code = curl_getinfo($curl, CURLINFO_HTTP_CODE);

    if ($status_code == 200)
    {
        // Read all template texts
        $templateText = file_get_contents($_FILES["fileTemplate"]["tmp_name"]);

        // 3. PARSE UPLOADED PDF DOCUMENT
        ParseDocument($apiKey, $uploadedFileUrl, $templateText);
    }
    else
    {
        // Display request error
        echo "<p>Status code: " . $status_code . "</p>";
        echo "<p>" . $result . "</p>";
    }
}
else
{
    // Display CURL error
    echo "Error: " . curl_error($curl);
}
}
else
{
    // Display service reported error
    echo "<p>Status code: " . $status_code . "</p>";
    echo "<p>" . $result . "</p>";
}
}

curl_close($curl);
}
else
{
    // Display CURL error
    echo "Error: " . curl_error($curl);
}
}

function ParseDocument($apiKey, $uploadedFileUrl, $templateText)
{
    // (!) Make asynchronous job
    $async = TRUE;

    // Prepare URL for Document parser API call.
    // See documentation: https://apidocs.pdf.co/#1-pdfdocumentparser
    $url = "https://api.pdf.co/v1/pdf/documentparser" .
        "?async=" . $async;

    // Post fields
    $data = array('url'=>$uploadedFileUrl, 'template'=>$templateText);

    // Create request
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_HTTPHEADER, array("x-api-key: " . $apiKey));
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_POST, true);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $data);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);

    // Execute request
    $result = curl_exec($curl);
    echo $result . "<br/>";

    if (curl_errno($curl) == 0)
    {
        $status_code = curl_getinfo($curl, CURLINFO_HTTP_CODE);

        if ($status_code == 200)
        {
            $json = json_decode($result, true);

            if ($json["error"] == false)

```

```

{
    // URL of generated JSON file that will available after the job completion
    $resultFileUrl = $json["url"];
    // Asynchronous job ID
    $jobId = $json["jobId"];

    // Check the job status in a loop
    do
    {
        $status = CheckJobStatus($jobId, $apiKey); // Possible statuses: "working", "failed", "aborted", "success"

        // Display timestamp and status (for demo purposes)
        echo "<p>" . date(DATE_RFC2822) . " : " . $status . "</p>";

        if ($status == "success")
        {
            // Display link to JSON file with information about parsed fields
            echo "<div><h2>Parsing Result:</h2><a href=\"" . $resultFileUrl . "\" target='_blank'> . $resultFileUrl . "</div>";
            break;
        }
        else if ($status == "working")
        {
            // Pause for a few seconds
            sleep(3);
        }
        else
        {
            echo $status . "<br/>";
            break;
        }
    }
    while (true);
}
else
{
    // Display service reported error
    echo "<p>Error: " . $json["message"] . "</p>";
}
}
else
{
    // Display request error
    echo "<p>Status code: " . $status_code . "</p>";
    echo "<p>" . $result . "</p>";
}
}
else
{
    // Display CURL error
    echo "Error: " . curl_error($curl);
}
}

function CheckJobStatus($jobId, $apiKey)
{
    $status = null;

    // Create URL
    $url = "https://api.pdf.co/v1/job/check?jobid=" . $jobId;

    // Create request
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_HTTPHEADER, array("x-api-key: " . $apiKey));
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);

    // Execute request
    $result = curl_exec($curl);

    if (curl_errno($curl) == 0)
    {
        $status_code = curl_getinfo($curl, CURLINFO_HTTP_CODE);

        if ($status_code == 200)
        {
            $json = json_decode($result, true);

            if ($json["error"] == false)

```

```
{
    $status = $json["status"];
}
else
{
    // Display service reported error
    echo "<p>Error: " . $json["message"] . "</p>";
}
}
else
{
    // Display request error
    echo "<p>Status code: " . $status_code . "</p>";
    echo "<p>" . $result . "</p>";
}
}
}
else
{
    // Display CURL error
    echo "Error: " . curl_error($curl);
}
}

// Cleanup
curl_close($curl);

return $status;
}
?>

</body>
</html>
```

---

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)