

www.bytescout.com

How to parse simple document for document parser API in PHP using PDF.co Web API

PDF.co Web API is the Rest API that provides set of data extraction functions, tools for documents manipulation, splitting and merging of pdf files. Includes built-in OCR, images recognition, can generate and read barcodes from images, scans 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:

AmazonAWS.yml

```
---
templateVersion: 3
templatePriority: 0
sourceId: Amazon Web Services Invoice
detectionRules:
  keywords:
  - Amazon Web Services
  - ATTN
  - Invoice
fields:
  total:
    type: macros
    expression: 'TOTAL AMOUNT DUE ON{{Anything}}{{Dollar}}{{Number}}'
    dataType: decimal
  subTotal:
    type: macros
```

```

expression: '{{LineStart}}{{Spaces}}Charges{{Spaces}}{{Dollar}}>{{Number}}'
dataType: decimal
dateIssued:
  type: macros
  expression: 'Invoice Date:{{Spaces}}>{{Anything}}{{LineEnd}}'
  dataType: date
  dateFormat: MMMM d , yyyy
invoiceId:
  type: macros
  expression: 'Invoice Number:{{Spaces}}>{{Digits}}'
companyName:
  type: static
  expression: Amazon Web Services, Inc.
companyWebsite:
  type: static
  expression: aws.amazon.com
billTo:
  type: rectangle
  expression: 'Bill to Address:{{ToggleSingleLineMode}}>{{AnythingGreedy}}'
  rectangle:
    - 33
    - 115.5
    - 213.75
    - 72.75
pageIndex: 0
currency:
  type: static
  expression: USD
tables:
- name: table1
  start:
    expression: '{{LineStart}}{{Spaces}}Detail{{LineEnd}}'
  end:
    expression: '{{EndOfPage}}'
  row:
    expression: '{{LineStart}}{{Spaces}}(?<description>{{SentenceWithSingleSpaces}}){{Spaces}}{{Dollar}}(?<unitPrice>'
  columns:
    - name: unitPrice
      type: decimal

```

DigitalOcean.yml

```

---
templateVersion: 3
templatePriority: 0
sourceId: DigitalOcean Invoice
detectionRules:
  keywords:
    # Template will match documents containing the following phrases:
    - DigitalOcean
    - 101 Avenue of the Americas
    - Invoice Number
fields:
  # Static field that will "DigitalOcean" to the result
  companyName:
    type: static
    expression: DigitalOcean
  # Macro field that will find the text "Invoice Number: 1234567" and return "1234567" to the result
  invoiceId:
    type: macros
    expression: 'Invoice Number: ({{Digits}})'
  # Macro field that will find the text "Date Issued: February 1, 2016" and return the date "February 1, 2016" in ISO form
  dateIssued:
    type: macros
    expression: 'Date Issued: ({{SmartDate}})'

```

```
dataType: date
dateFormat: auto-mdy
# Macro field that will find the text "Total:
```

```
{codeFileName}
```

```
{code}
```

```
10.00" and return "110.00" to the result
```

```
total:
  type: macros
  expression: 'Total: {{Dollar}}{{Number}}'
  dataType: decimal
# Static field that will "USD" to the result
currency:
  type: static
  expression: USD
```

```
tables:
```

```
- name: table1
# The table will start after the text "Description    Hours"
start:
  expression: 'Description{{Spaces}}Hours'
# The table will end before the text "Total:"
end:
  expression: 'Total:'
# Macro expression that will find table rows "Website-Dev (1GB) 744 01-01 00:00 01-31 23:59
```

```
{codeFileName}
```

```
{code}
```

```
0.00", etc.
```

```
row:
# Groups <description>, <hours>, <start>, <end> and <unitPrice> will become columns in the result table.
  expression: '{{LineStart}}{{Spaces}}?<description>{{SentenceWithSingleSpaces}}{{Spaces}}(?<hours>{{Digits}}){{S
# Suggest data types for table columns (missing columns will have the default "string" type):
columns:
- name: hours
  type: integer
- name: unitPrice
  type: decimal
```

```

---
templateVersion: 3
templatePriority: 0
sourceId: Google Invoice
detectionRules:
  keywords:
    - Google
    - 77-0493581
    - Invoice
fields:
  invoiceId:
    expression: 'Invoice number:{{Spaces}}{{Digits}}'
  dateIssued:
    expression: 'Issue date:{{Spaces}}{{SmartDate}}'
    dataType: date
    dateFormat: MMM d, yyyy
  total:
    expression: 'Amount due in USD:{{Spaces}}{{Number}}'
    dataType: decimal
  subTotal:
    expression: 'Subtotal in USD:{{Spaces}}{{Number}}'
    dataType: decimal
  taxRate:
    expression: 'State sales tax {{OpeningParenthesis}}{{Digits}}{{Percent}}{{ClosingParenthesis}}'
    dataType: integer
  tax:
    expression: 'State sales tax{{Anything}}{{Number}}{{LineEnd}}'
    dataType: decimal
  companyName:
    type: static
    expression: 'Google LLC'
  billTo:
    type: rectangle
    rectangle:
      - 0
      - 152
      - 280
      - 72
    pageIndex: 0
  billingId:
    expression: 'Billing ID:{{Spaces}}{{DigitsOrSymbols}}'
  currency:
    type: static
    expression: 'USD'
tables:
- name: table1
  start:
    expression: 'Description{{Spaces}}Interval{{Spaces}}Quantity{{Spaces}}Amount'
  end:
    expression: 'Subtotal in USD'
  row:
    expression: '{{LineStart}}{{Spaces}}(?<description>{{SentenceWithSingleSpaces}}){{Spaces}}(?<interval>{{3Letters}}'
  columns:
    - name: quantity
      type: integer
    - name: amount
      type: decimal

```

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'>" . $resultFileUrl . "</a>";
                        break;
                    }
                    else if ($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