Recommended way to embed PDF in HTML? [closed]

Asked 17 years, 1 month ago Modified 10 months ago Viewed 2.0m times

1398

Closed. This question is opinion-based. It is not currently accepting answers.

Closed 1 year ago.

This question's answers are a community effort. Edit existing answers to improve this post. It is not currently accepting new answers or interactions.

What is the recommended way to embed PDF in HTML?

- iFrame?
- Object?
- Embed?

What does Adobe say itself about it?

In my case, the PDF is generated on the fly, so it can't be uploaded to a third-party solution prior to flushing it.

html pdf

Share Follow

edited Oct 6, 2012 at 12:28

community wiki
Daniel Silveira

9 Try pdf2htmlEX: github.com/coolwanglu/pdf2htmlEX – Lu Wang May 10, 2013 at 9:20

1 There's a great comparison not just of specific solutions but general strategies, on the pdf2htmlEX repo's wiki. Also, though I haven't tried it, this seems to be a maintained fork. – ShreevatsaR Apr 23, 2019 at 19:51

Try approach described in sitelint.com/blog/... – Cezary Tomczyk Oct 18, 2023 at 17:08

Comments disabled on deleted / locked posts / reviews

24 Answers

Sorted by: Highest score (default)

710

This is quick, easy, to the point and doesn't require any third-party script:

Copy

```
<embed src="http://example.com/the.pdf" width="500" height="375"
 type="application/pdf">
```

UPDATE (2/3/2021)

Adobe now offers its own PDF Embed API.
(That requires registering at Adobe and get clientID to use within js)

https://www.adobe.io/apis/documentcloud/dcsdk/pdf-embed.html

UPDATE (1/2018):

The Chrome browser on Android no longer supports PDF embeds. You can get around this by using the Google Drive PDF viewer

Copy

```
<embed src="https://drive.google.com/viewerng/
viewer?embedded=true&url=http://example.com/the.pdf" width="500" height="375">
```

Share Improve this answer Follow

edited Apr 6, 2023 at 15:56

community wiki
Batfan

Sign up to request clarification or add additional context in comments. ✕

7 Comments

Add a comment

Jonathon Hill Over a year ago

Better to use an <object> tag so that you can include a fall-back.

▲ 41 Reply ···

Hassek Over a year ago

if you want to get sure it will be shown instead of auto download the pdf (as it happend to me) add `type='application/pdf'` to the embed tag

▲ 76 Reply ···

Raphael Over a year ago

Better to use `<object>` tag because it can display alternate content in browsers that can't display pdfs. Can even put an `<embed>` tag in an `<object>` tag if you want. Ref: stackoverflow.com/questions/1244788/embed-vs-object

▲ 41 Reply ···

CodeForGood Over a year ago

@batfan, is it correct if I suppose nearly all solutions offered here address the challenges of displaying PDF document perfectly on screens other than mobiles? I have a webpage where the PDF is embedded and this works fine on the computer screens, but doesn't on the android phones that use Chrome. Please advise some tips/solution to this issue.

▲ 0 Reply ···

Batfan Over a year ago

@PKTG I just tested this and it's working for me. See below jsfiddle.net/196fv04s

▲ 2 Reply ···

|

▲

611

▼

Probably the best approach is to use the PDF.JS library. It's a pure HTML5/JavaScript renderer for PDF documents without any third-party plugins.

Online demo: https://mozilla.github.io/pdf.js/web/viewer.html

GitHub: https://github.com/mozilla/pdf.js

Share Improve this answer Follow

edited May 14, 2021 at 2:27

community wiki
lubos hasko

6 Comments

Add a comment

Peter Craig Over a year ago

As noted in another answer, scribd actually uses pdf2swf to convert pdf files

▲ 16 Reply ···

frankster Over a year ago

I highly recommend against using scribd - I have just performed an experiment on a particular document and in firefox 4 it only displays the first 3 pages, whereas in IE9 its rendering text wrong - its offset some sections of the page. So technically speaking it is buggy. Additionally they expect you to subscribe to print or download documents! Essentially they are taking previously free documents and erecting a paywall around them.

LarsH Over a year ago

PDF.js library actually looks like a very good solution, although the linked demo doesn't show it embedded in a page (it takes up the whole page). But it uses HMTL5 canvas, so it should be easy to embed, and it's *fast*. On the down side, it takes some js to use, unlike `<object>` : github.com/mozilla/pdf.js/blob/master/examples/helloworld/...

▲ 12 Reply ···

Rocco The Taco Over a year ago

pdf.js works poorly on tablet browsers such as Chrome. I also find it to be very, very slow for larger pdf documents.

▲ 46 Reply ···

Kiquenet Over a year ago

@RoccoTheTaco any alternative with high performance?

▲ 2 Reply ···

|

▲

385

▼

You can also use Google PDF viewer for this purpose. As far as I know it's not an official Google feature (am I wrong on this?), but it works for me very nicely and smoothly. You need to upload your PDF somewhere before and just use its URL:

Copy

```
<iframe src="https://docs.google.com/gview?
url=http://example.com/mypdf.pdf&embedded=true" style="width:718px;
height:700px;" frameborder="0"></iframe>
```

What is important is that it doesn't need a Flash player, it uses JavaScript.

Share Improve this answer Follow

edited Oct 8, 2022 at 17:50

community wiki
Lukasz

5 Comments

Add a comment

SuperElectric Over a year ago

One thing that should be mentioned is that there's an upper limit to the size of the PDF that can be displayed. I think it's currently 10MB/100 pages. PS: I don't think the viewer is "unofficial"; they even have a how-to page that constructs the embed URL for you: docs.google.com/viewer

▲ 35 Reply ···

KoviNET Over a year ago

This option works great, but you have to make your PDFs publicly accessible, which for me is not always an option.

▲ 34 Reply ···

dwenaus Over a year ago

@riot_starter this solution no longer works, if you are fully logged out of google it loads fine, if you are fully logged in it loads fine, however if you are in between (whatever that means) it asks you for your password. And in the case of the iframe it just does not load.

▲ 6 Reply ···

arkon Over a year ago

You are allowed to set the privacy of any Google Docs between `public` , `sign-in required` , and `private` . It is definitely an official feature, considering any document on Google Docs has an `embed` option.

▲ 5 Reply ···

kommradHomer Over a year ago

Another thing that should be mentioned is the "you have made too many requests recently" warning , which is a limitation.

▲ 7 Reply ···

▲

147

You do have some control over how the PDF appears in the browser by passing some options in the query string. I was happy to this working, until I realized it does not work in IE8. :(

It works in Chrome 9 and Firefox 3.6, but in IE8 it shows the message "Insert your error message here, if the PDF cannot be displayed."

I haven't yet tested older versions of any of the above browsers, though. But here's the code I have anyway in case it helps anyone. This sets the zoom to 85%, removes scrollbars, toolbars and nav panes. I'll update my post if I do come across something that works in IE as well.

Copy

```
<object width="400" height="500" type="application/pdf" data="/my_pdf.pdf?#zoom=85&scrollbar=0&toolbar=0&navpanes=0">
    <p>Insert your error message here, if the PDF cannot be displayed.</p>
</object>
```

Share Improve this answer Follow

edited Aug 17, 2018 at 7:59

community wiki
Gayle

3 Comments

Add a comment

Adrian P. Over a year ago
That's the best solution because is using the browser capabilities and not a complicated third-party solution. In all modern browsers (IE9, FF or Chrome) PDF should embed nicely. Sorry for IE 6/7 users. They have to upgrade. We've stop supporting these browsers long time ago. :(

12 Reply ···

gojiraki Over a year ago
This solution works best. I have tried other solutions from stackoverflow but I could not get to display pdf. This is the best solution.

0 Reply ···

avijit Over a year ago
I am trying to show pdf file form s3 presigned URL. My file types are set as pdf in S3. Unfortunately embed, ojbect, iframe all triggers download without showing the pdf.

0 Reply ···

75

Using both `<object>` and `<embed>` will give you a wider breadth of browser compatibility.

Copy

```
<object data="http://yoursite.com/the.pdf" type="application/pdf" width="750px" height="750px">
    <embed src="http://yoursite.com/the.pdf" type="application/pdf">
        <p>This browser does not support PDFs. Please download the PDF to view it: <a href="http://yoursite.com/the.pdf">Download PDF</a>.</p>
    </embed>
</object>
```

Share Improve this answer Follow

edited Aug 8, 2017 at 10:55

community wiki
Suneel Kumar

5 Comments

Add a comment

Umair Hamid Over a year ago
This will not work in browsers if pdf addon in not installed or out of date

6 Reply ···

NotJay Over a year ago
Also, this will not validate for those that are concerned with code validation.

1 Reply ···

Richie Thomas Over a year ago

This worked great for me, whereas using the `embed` tag by itself was deemed `unsafe` by Chrome and Firefox.

▲ 1 Reply •••

Ivan Rubinson Over a year ago

For some reason on Android (chrome) it doesn't embed the PDF and instead offers to download it. On iPhone (safari) it shows only the first page of the PDF.

▲ 3 Reply •••

Prabhu Thomas Over a year ago

@IvanRubinson Android on phones and on some tablets does not have the capability to display embedded pdfs.. also for some iOS mobile devices (iPhone, iPad).. on the other iOS devices (iPad Pro, ..) only the first page of the pdf will be shown.. this is a known issue on iOS which has been there for a very long time..

▲ 0 Reply •••

▲
25
▼

Have a look for this code- To embed the PDF in HTML

Copy

```
<!-- Embed PDF File -->
<object src="YourFile.pdf" type="application/pdf" title="SamplePdf" width="500"
height="720">
    <a href="YourFile.pdf">shree</a>
</object>
```

Share Improve this answer Follow

edited Nov 2, 2021 at 8:23

community wiki
5 revs, 5 users 52%
amIT

4 Comments ⌄

Add a comment

Mottie Over a year ago

This doesn't seem to work. I always end up with a blank grey screen instead of the pdf. But switching `src` for `data`, as in @Gayle's answer, does work.

▲ 10 Reply •••

NotJay Over a year ago

I am using Chrome 44 and it won't work. I also tried switching data to src.

▲ 1 Reply •••

Gavin Simpson Over a year ago

type="application/pdf" fixes the problem. Works fine in chrome.

▲ 2 Reply •••

Jared Wilber Over a year ago

Including type="application/x-pdf" will now actually prevent rendering on both Chrome and FireFox. It should be removed.

▲ 2 Reply •••

▲
22
▼

Convert it to PNG via ImageMagick, and display the PNG (quick and dirty).

Copy

```
<?php
  $dir = '/absolute/path/to/my/directory/';
  $name = 'myPDF.pdf';
  exec("/bin/convert $dir$name $dir$name.png");
  print '<img src="$dir$name.png" />';
?>
```

This is a good option if you need a quick solution, want to avoid cross-browser PDF viewing problems, and if the PDF is only a page or two. Of course, you need ImageMagick installed (which in turn needs Ghostscript) on your web server, an option that might not be available in shared hosting environments. There is also a PHP plugin (called imagick) that works like this but it has its own special requirements.

Share Improve this answer Follow

edited Jul 4, 2022 at 2:49

community wiki
Dan Mantyla

9 Comments

Add a comment

Garrett Over a year ago
Or convert the image to PNG using an application, such as Preview on Mac.

▲ 1 Reply •••

Dan Mantyla Over a year ago
but can you do that programmatically?

▲ 2 Reply •••

seb Over a year ago
what if the document has more than 1 page? I guess you will only show the first page then

▲ 0 Reply •••

99 Problems - Syntax ain't one Over a year ago
If you open the PDF using the zend library then you can do a `foreach($pdf->pages as $page)` and create an image for each page

▲ 2 Reply •••

Charles Duffy Over a year ago
(though something that was true even twelve years ago is that there's security impact here: You wouldn't want a file named `$(rm -rf ~).pdf` -- and yes, that's a valid filename -- to be converted this way; it's much safer to use an exec form that takes a list/array of arguments rather than a string that requires shell parsing)

▲ 1 Reply •••

|

▲ 14 ▼

You can use the relative location of the saved pdf like this:

Example1

Copy

```
<embed src="example.pdf" width="1000" height="800" frameborder="0"
allowfullscreen>
```

Example2

Copy

```
<iframe src="example.pdf" style="width:1000px; height:800px;" frameborder="0"
allowfullscreen></iframe>
```

Share Improve this answer Follow

edited Dec 17, 2020 at 1:02

community wiki
Ahmedakhtar11

1 Comment

Add a comment

Mehmet Taha Meral Over a year ago
Chrome understands that embed is like Flash and it automatically blocks it. iFrame with bootstrap is a good option.

▲ 1 Reply •••

12

1. Create a container to hold your PDF

```
<div id="example1"></div>
```

2. Tell PDFObject which PDF to embed, and where to embed it

```
<script src="/js/pdfobject.js"></script>
<script>PDFObject.embed("/pdf/sample-3pp.pdf", "#example1");</script>
```

3. You can optionally use CSS to specify visual styling, including dimensions, border, margins, etc.

```
<style>
.pdfobject-container { height: 500px;}
.pdfobject { border: 1px solid #666; }
</style>
```

source : https://pdfobject.com/

Share Improve this answer Follow

edited Sep 30, 2019 at 20:55

community wiki
Said Bouigherdaine

3 Comments

Add a comment

netfed Over a year ago
This was a very good solution and worked well in conjunction with bootstrap styling. It all went on top of other bootstrap elements.

0 Reply ···

Hoang Subin Over a year ago
Did you try with blob file? I used this with `pdf.output('bloburl')` from lib jspdf but it seems not working on IE11.

0 Reply ···

Mehmet Taha Meral Over a year ago
This is a good solution but it doesn't work even the latest iPhone safari browser :(

0 Reply ···

12

Our problem is that for legal reasons **we are not allowed to temporarily store a PDF on the hard disk**. In addition, the entire page should not be reloaded when displaying a PDF as Preview in the Browser.

First we tried PDF.jS. It worked with Base64 in the viewer for Firefox and Chrome. However, it was unacceptably slow for our PDF. IE/Edge didn't work at all.

We therefore tried it with a Base64 string in an HTML object tag. This again didn't work for IE/Edge (maybe the same problem as with PDF.js). In Chrome/Firefox/Safari again no problem. That's why we chose a hybrid solution. **Edge we use an IFrame and for all other browsers the object-tag.**

The IFrame solution would of course also work for Chrome and co. The reason why we didn't use this solution for Chrome is that although the PDF is displayed correctly, *Chrome makes a new request to the server as soon as you click on "download" in the preview*. The required hidden-field *pdfHelperTransferData* (for sending our form data needed for PDF generation) is no longer set because the PDF is displayed in an IFrame. For this feature/bug see Chrome sends two requests when downloading a PDF (and cancels one of them).

Now the problem children IE9 and IE10 remain. For these we gave up a preview solution and simply send the document by clicking the preview button as a download to the user (instead of the preview). We have tried a lot but even if we had found a solution the extra effort for this tiny part of users would not have been worth the effort. You can find our solution for the download here: Download PDF without refresh with IFrame.

Our Javascript:

Copy

```
var transferData = getFormAsJson()
if (isMicrosoftBrowser()) {
        // Case IE / Edge (because doesn't recoginzie Pdf-Base64 use Iframe)
        var form = document.getElementById('pdf-helper-form');
        $("#pdfHelperTransferData").val(transferData);
        form.target = "iframe-pdf-shower";
        form.action = "serverSideFunctonWhichWritesPdfinResponse";
        form.submit();
 } else {
        // Case non IE use Object tag instead of iframe
        $.ajax({
            url: "serverSideFunctonWhichRetrivesPdfAsBase64",
            type: "post",
            data: { downloadHelperTransferData: transferData },
            success: function (result) {
                $("#object-pdf-shower").attr("data", result);
            }
        })
 }
```

Our HTML:

Copy

```
<div id="pdf-helper-hidden-container" style="display:none">
   <form id="pdf-helper-form" method="post">
        <input type="hidden" name="pdfHelperTransferData"
id="pdfHelperTransferData" />
   </form>
</div>

<div id="pdf-wrapper" class="modal-content">
    <iframe id="iframe-pdf-shower" name="iframe-pdf-shower"></iframe>
    <object id="object-pdf-shower" type="application/pdf"></object>
</div>
```

To check the browser type for IE/Edge see here: [How can I detect Internet Explorer (IE) and Microsoft Edge using JavaScript?]() I hope these findings will save someone else the time.

Share Improve this answer Follow

edited Jun 5, 2022 at 10:36

community wiki
Astrophage

2 Comments

Add a comment

Kate Over a year ago

I'm sorry, but i don't understand, what kind of value does variable "transferData " have?

▲ 0 Reply ⋯

Astrophage Over a year ago

Hi @Kate, In our case, we generate the PDF based on a user's input in our web form. "transferData" contains the data which are sent as JSON to the PDF generation function in the web API. I've added a line more code for clarification.

▲ 1 Reply ⋯

▲
11
▼

[Update] Flash is no longer allowed to run in modern devices. (Dead redirected link removed)

FORMER `FDView` combined [PDF2SWF]() (which itself was based on [xpdf]()) with an [SWF]() viewer so you can convert and embed PDF documents on the fly on your server.

xpdf is ~~not~~ a perfect PDF converter. If you needed better results in the past then Older GhostGum unsupported GhostView 5 spinoff was a possibility, currently ***Ghostscript 10 is an interpreter, not a viewer. As such it offers almost no UI.*** But can view PDF etc offline.

GhostScript does have multiple abilities to convert PDF documents into other formats which you may be able to more easily build a ~~Flash~~ **HTML** viewer for. [NOTE PDF.js is based on similar convert pages into images of a PDF.]

But for simple PDF documents, FDView [at the time] did work reasonably well.

Share Improve this answer Follow

edited Feb 10, 2025 at 13:50

community wiki
4 revs, 3 users 56%
Adam Davis

2 Comments

Add a comment

Michael Myers Over a year ago
Is FDView available anywhere else? code4net.com seems to have disappeared.

0 Reply •••

Adam Davis Over a year ago
@Michael Not that I can easily find. I'll leave this answer up on the off chance that someone re-hosts fdview somewhere else.

0 Reply •••

5

This is the way I did with **AXIOS** and Vue.js:

Copy

```
        axios({
            url: `urltoPDFfile.pdf`,
            method: 'GET',
            headers: headers,
            responseType: 'blob'
        })
        .then((response) => {
            this.urlPdf = URL.createObjectURL(response.data)
        })
        .catch((error) => {
            console.log('ERROR   ', error)
        })
```

add urlPDF dynamically to HTML:

Copy

```
<object width='100%' height='600px' :data='urlPdf' type='application/pdf'>
</object>
```

Share Improve this answer Follow

answered May 17, 2018 at 9:26

community wiki
Despertaweb

Comments

Add a comment

5

Update - Adobe PDF Embed API

Adobe released their Adobe PDF Embed API which is completely free. Since they created the PDF format itself, their API is probably the best for this.

- It delivers the highest quality PDF rendering available.
- You can fully customize user experience and choose how to display a PDF.
- You will also have analytics on PDF usage so you can understand how users interact with PDFs, including time spent on a page and searches.

All you have to do is create an `api_key` and use it in the code snippet.

Displaying PDF by `file_url`

Here is the example of the code snippet that you can just add to your HTML and take advantage of their API for displaying PDF by `file_url`. You would have to add `{ location: { url: "url_of_the_pdf" } }` config.

Copy

```
<div id="adobe-dc-view"></div>

<script src="https://documentcloud.adobe.com/view-sdk/main.js"></script>

<script type="text/javascript">
  document.addEventListener("adobe_dc_view_sdk.ready", function(){
    var adobeDCView = new AdobeDC.View({clientId: "api_key", divId: "adobe-dc-view"});
    adobeDCView.previewFile({
      content: { location: { url: "url_of_the_pdf" } },
      metaData: { fileName: "file_name_to_display" }
    }, {});
  });
</script>
```

Displaying PDF as buffer

Here is the example of the code snippet that you can just add to your HTML and take advantage of their API for displaying PDF if you have the buffer (local file for example). You would have to add `{ promise: <FILE_PROMISE> }` config.

Copy

```
<div id="adobe-dc-view"></div>

<script src="https://documentcloud.adobe.com/view-sdk/main.js"></script>

<script type="text/javascript">
  document.addEventListener("adobe_dc_view_sdk.ready", function(){
    var adobeDCView = new AdobeDC.View({clientId: "api_key", divId: "adobe-dc-view"});
    adobeDCView.previewFile({
      content: { promise: <FILE_PROMISE> }
      metaData: { fileName: "file_name_to_display" }
    }, {});
  });
</script>
```

Share Improve this answer Follow

edited Jan 17, 2022 at 11:41

community wiki
NeNaD

Comments

Add a comment

3

PdfToImageServlet using ImageMagick's `convert` command.

Usage example: `<img src='/webAppDirectory/PdfToImageServlet?pdfFile=/usr/share/cups/data/default-testpage.pdf'>`

Share Improve this answer Follow

answered Feb 23, 2017 at 20:09

community wiki
PLA

Comments

Add a comment

3

I had to preview a PDF with React so after trying several libraries my optimal solution was to fetch the data and ebmed it.

Copy

```
const pdfBase64 = //fetched from url or generated with jspdf or other library

  <embed
    src={pdfBase64}
    width="500"
    height="375"
    type="application/pdf"
  ></embed>
```

Share Improve this answer Follow

answered Dec 18, 2019 at 21:37

community wiki
Pecata

Comments

Add a comment

1

Copy

```
<object width="400" height="400" data="helloworld.pdf"></object>
```

Share Improve this answer Follow

answered Feb 3, 2016 at 14:36

community wiki
Daryl H

Comments

Add a comment

1

1. Construct a blob of the input PDF bytes
2. Use an **iframe** and **PDF.js** patched with this cross browser workaround

The URI for the iframe should look something like this:

Copy

```
/viewer.html?file=blob:19B579EA-5217-41C6-96E4-5D8DF5A5C70B
```

Now FF, Chrome, IE 11, and Edge all display the PDF in a viewer in the iframe passed via standard blob URI in the URL.

Share Improve this answer Follow

edited Jul 18, 2018 at 13:23

community wiki
fartwhif

Comments

Add a comment

1

[Update] seems to have been Branched into different payable apps (many tagged Kami) the google Play store version of Notable PDF is here https://play.google.com/store/apps/details?id=com.albankurti.notablepdf&hl=en_US

One of the options you should consider is **Notable PDF**
It has a free plan unless you are planning on doing real-time online collaboration on pdfs

Embed the following `iframe` to any html and enjoy the results:

Copy

```
<iframe width='1000' height='800'
src='https://play.google.com/store/apps/details?
id=com.albankurti.notablepdf&hl=en_US' frameborder='0' allowfullscreen>
</iframe>
```

Share Improve this answer Follow

edited Aug 2, 2024 at 1:53

community wiki
2 revs, 2 users 80%
George Pligoropoulos

Comments

Add a comment

-1

To stream the file to the browser, see Stack Overflow question *How to stream a PDF file as binary to the browser using .NET 2.0* - note that, with minor variations, this should work whether you're serving up a file from the file system or dynamically generated.

With that said, the referenced MSDN article takes a rather simplistic view of the world, so you may want to read *Successfully Stream a PDF to browser through HTTPS* as well for some of the headers you may need to supply.

Using that approach, an iframe is probably the best way to go. Have one webform that streams the file, and then put the iframe on another page with its `src` attribute set to the first form.

Share Improve this answer Follow

edited May 23, 2017 at 12:26

community wiki
GalacticCowboy

Comments

Add a comment

-1

I found that the best way to embed a pdf for my case was by using bootstrap because not only does it show the pdf but it also fill available space and you can specify the ratio as you wish. Here's an example of what i made with it:

Copy

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">

<div class="embed-responsive embed-responsive-1by1">
  <iframe class="embed-responsive-item" src="http://example.com/the.pdf"
type="application/pdf" allowfullscreen></iframe>
</div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js">
```

```
</script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js">
</script>
```

▶ Run code snippet Expand

Share Improve this answer Follow

answered Aug 15, 2020 at 18:31

community wiki
Marc Partensky

Comments

Add a comment

-1

PDF.js is still the best for me. I use `pdfjs-viewer-element` to simplify PDF.js setup:

Copy

```
<script type="module" src="https://cdn.skypack.dev/pdfjs-viewer-element">
</script>
```

Specify the PDF.js release directory with `viewer-path` and path to PDF file with `src` attributes:

Copy

```
<pdfjs-viewer-element
  id="viewer"
  src="/file.pdf"
  viewer-path="/pdfjs-4.0.379-dist"
  style="height: 600px">
</pdfjs-viewer-element>
```

More samples for embedding PDF.js https://github.com/alekswebnet/pdfjs-viewer-element/tree/master/demo

Share Improve this answer Follow

edited Mar 29, 2024 at 0:15

community wiki
alekswebnet

Comments

Add a comment

-2

Before I got a problem with embeding base64 encoded with PDF because the URI limitation, so any files over 2MB won't render properly on Chrome.

My solution is:

1. Convert uri encoded to Blob:
2. Generate the temporary DOM String base on Blob.

   `const blob = dataURItoBlob(this.dataUrl);`

   `var temp_url = window.URL.createObjectURL(blob);`

3. Decide where you want to attach the iframe to DOM:

   `const target = document.querySelector(targetID);`

   `` target.innerHTML = `<iframe src='${temp_url}' type="application/pdf"></iframe> ``

Share Improve this answer Follow

answered Aug 18, 2020 at 3:35

community wiki
Tony Bui

Comments

Add a comment

-2

If you don't want to host the PDFs yourself or want to customize your PDF viewer with additional security features like preventing users to download the PDF file. I recommend using CloudPDF. https://cloudpdf.io

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CloudPDF Viewer</title>
  <style>
    body, html {
      height: 100%;
      margin: 0px;
    }
  </style>
</head>
<body style="height: 100%">
  <div id="viewer" style="width: 800px; height: 500px; margin: 80px auto;">
</div>
  <script type="text/javascript" src="https://cloudpdf.io/viewer.min.js?
version=0.1.0-beta.11"></script>
  <script>
    document.addEventListener('DOMContentLoaded', function(){
      const config = {
        documentId: 'eee2079d-b0b6-4267-9812-b6b9eadb9c60',
        darkMode: true,
      };
      CloudPDF(config, document.getElementById('viewer')).then((instance) => {

      });
    });
  </script>
 </body>
</html>
```

Share Improve this answer Follow

edited Feb 8, 2021 at 20:54

community wiki
Bob Singor

Comments

Add a comment

-2

Workaround in Firefox for the Toolbar.

I little preface ...

I need to have a couple of SVG's for the Layout to accomplish a Document in DIN-5008.

So I got me a copy of the Wikipedia SVG, that shows the Dimensions and the Placements. Additionally, I made some SVG's for the Logo and some drawings as a Watermark.

I then made a HTML5 File and in it, I placed one `div` to hold the SVG's and the PDF. All of them are placed into a separate `div` each.

Then I used CSS-Styles (classes) to align them over each other. I made a class Transparent_025.

```
.Transparent_025 { opacity: 0.25; }
```

With that, I can Stack the SVG's and the PDF to see all align properly.

But Firefox doesn't use the `toolbar=1|0`.

```
<div class="SVGStack_A Transparent_025" style="position: absolute; top:
-32px;">
    <object
        data="thePath/theFilename.pdf?
#zoom=100%&scrollbar=0&toolbar=0&navpanes=0"
        type="image/svg+xml" width="793.7px" height="1122.52px" />
</div>
```

This is my Workaround.

So the **Height** of the Toolbar in Firefox is ***32px***. And keep the PDF **Scale to 100%**.

PS: The Documents are in DIN-A4, so I used this:

```
<meta name="viewport" content="width=210mm, initial-scale=1.0">
```

With that, a needed to set the Width and Height.

```
width="793.7px" height="1122.52px"
```

Also, all `div` are placed absolutely on the Lop-Left-Corner.

Be aware that the PDF-Viewer in Firefox can get the Focus and a Scroll or Zoom can happen.

Share Improve this answer Follow

edited Jul 6, 2024 at 3:35

community wiki
Zui

Comments

Add a comment

Start asking to get answers

Find the answer to your question by asking.

Ask question

Explore related questions

html pdf

See similar questions with these tags.