How to display a PDF as an image in React app using URL?

Last Updated : 13 Aug, 2020

If we use the fetch method then it will open a new window for displaying the PDF file and let users download the PDF. But if you don't want that then there is a way to do so. You can use the package called **react-pdf,** by using this package you can render the PDF in your React app by using the PDF URL.

Prerequisites:

- Your project needs to use React 16.3 or later.
- Basic knowledge of packages

React-pdf: It lets you display PDF in your React app as easily as if they were images. It helps to create custom components that you can use to create and structure your PDF documents.

Step 1: Create React App

```
npx create-react-app appname

cd appname
```

```
npm start
```

Step 2: Install react-pdf package.

```
npm install react-pdf
```

Step 3: First make a **separate component** PDF (name of the component, can be anything) and render the PDF component in App.js.
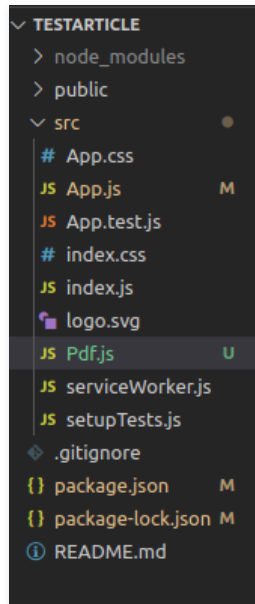
App.js:

Javascript

```javascript
import React from 'react';
import Pdf from './Pdf'

const App = ()=> {

  return (
    <div className="App">
        //Rendering a pdf component
        <Pdf />
    </div>
  );
}
```

```
        export default App;
```

After creating a pdf component your project directory will look like this.

Step 4: In this section, we load the PDF and render it on your app.

Document: Loads a document passed using file prop.

File Prop: It tells what PDF should be displayed, In the above code, we pass URL to it.

URL: The URL consists of two parts here.

- The 1st part is due to preventing cors error, you may refer docs to read more about the core.

```
1st part: https://cors-anywhere.herokuapp.com/
```

- The 2nd part is our actual URL of PDF.

```
2nd part: http://www.pdf995.com/samples/pdf.pdf
```

One more thing we need to do is ENABLE **PDF.JS WORKER**, you could use pdf.worker.js from an external CDN.

onDocumentLoadSuccess: When the document gets successfully loaded we set the state of page number to tells on which page number of pdf the user is.

Pdf.js: Now open the PDF component.

Javascript

```javascript
import React, { useState } from 'react';
import { Document, Page,pdfjs } from 'react-pdf';
import './pdf.css'

//PDFjs worker from an external cdn
const url =
"https://cors-anywhere.herokuapp.com/http://www.pdf995.com/samples/pdf.pdf"

export default function Test() {

    pdfjs.GlobalWorkerOptions.workerSrc =
    `//cdnjs.cloudflare.com/ajax/libs/pdf.js/${pdfjs.version}/pdf.worker.js`;
     const [numPages, setNumPages] = useState(null);
```

```
      const [pageNumber, setPageNumber] = useState(1);

        function onDocumentLoadSuccess({ numPages }) {
        setNumPages(numPages);
        setPageNumber(1);
      }
      return (
        <>
        <div className="main">
          <Document
            file={url}
            onLoadSuccess={onDocumentLoadSuccess}
            >
            <Page pageNumber={pageNumber} />
          </Document>
         </div>
        </>
      );
    }
```

Step 5: Now the last thing add NEXT and PREVIOUS buttons to PDF file.

Pdf.js: Here we added two buttons **NEXT AND PREVIOUS** and their functions named **previousPage()** and **nextPage()** which change the state of the current page.

Javascript

```
import React, { useState } from 'react';
import { Document, Page,pdfjs } from 'react-pdf';
```

```
const url =
"https://cors-anywhere.herokuapp.com/http://www.pdf995.com/samples/pdf.pdf"

export default function Test() {

  pdfjs.GlobalWorkerOptions.workerSrc =
  `//cdnjs.cloudflare.com/ajax/libs/pdf.js/${pdfjs.version}/pdf.worker.js`;
  const [numPages, setNumPages] = useState(null);
  const [pageNumber, setPageNumber] = useState(1);

  /*To Prevent right click on screen*/
  document.addEventListener("contextmenu", (event) => {
    event.preventDefault();
  });

  /*When document gets loaded successfully*/
  function onDocumentLoadSuccess({ numPages }) {
    setNumPages(numPages);
    setPageNumber(1);
  }

  function changePage(offset) {
    setPageNumber(prevPageNumber => prevPageNumber + offset);
  }

  function previousPage() {
    changePage(-1);
  }

  function nextPage() {
    changePage(1);
  }

  return (
    <>
```
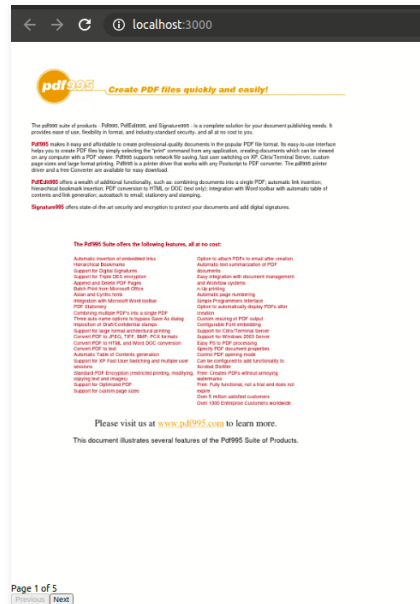
```jsx
        <div className="main">
          <Document
            file={url}
            onLoadSuccess={onDocumentLoadSuccess}
          >
            <Page pageNumber={pageNumber} />
          </Document>
          <div>
            <div className="pagec">
              Page {pageNumber || (numPages ? 1 : '--')} of {numPages || '--'}
            </div>
            <div className="buttonc">
            <button
              type="button"
              disabled={pageNumber <= 1}
              onClick={previousPage}
              className="Pre"

            >
              Previous
            </button>
            <button
              type="button"
              disabled={pageNumber >= numPages}
              onClick={nextPage}

            >
              Next
            </button>
            </div>
          </div>
        </div>
      </>
    );
  }
```

Output:

Like 0

RECOMMENDED ARTICLES

Page :

Article Contributed By :

SHOBHITGARG1

Vote for difficulty

Easy   Normal   Medium   Hard   Expert

Article Tags :   react-js, JavaScript, Web Technologies, Web technologies Questions

Improve Article   Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

WHAT'S NEW

Full Stack Development with React & Node JS - Live

View Details

Complete Interview Preparation

View Details

System Design-Live Classes for Working Professionals

MOST POPULAR IN JAVASCRIPT

How to calculate the number of days between two dates in javascript?

Convert a string to an integer in JavaScript

Difference between var and let in JavaScript

File uploading in React.js

Difference between TypeScript and JavaScript

MOST VISITED IN WEB TECHNOLOGIES

Installation of Node.js on Linux

Top 10 Projects For Beginners To Practice HTML and CSS Skills

How to insert spaces/tabs in text using HTML/CSS?

Web 1.0, Web 2.0 and Web 3.0 with their difference

How to calculate the number of days between two dates in javascript?

Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

About Us

Careers

Privacy Policy

Contact Us

Copyright Policy

Learn

Algorithms

Data Structures

Languages

CS Subjects

Video Tutorials

Web Development

Web Tutorials

HTML

CSS

JavaScript

Bootstrap

Contribute

Write an Article

Write Interview Experience

Internships

Videos

@geeksforgeeks , Some rights reserved

Create PDF in your applications with the Pdfcrowd HTML to PDF API

PDFCROWD