How Accusoft PDF Viewer Improves Upon PDF.js

The ability to view PDF files has become an essential feature for web-based applications. While dedicated desktop readers are still common, the average user justifiably expects to be able to view documents without switching between applications. Thanks to browser-based PDF libraries like PDF.js, developers can both integrate the viewing features they need and build [the next generation](https://www.accusoft.com/resources/blog/introducing-accusoft-pdf-viewer-a-free-javascript-pdf-library/) of PDF viewing integrations.

What Is PDF.js?

An open-source JavaScript PDF library, PDF.js was originally developed by the Mozilla Foundation in 2011 to serve as the built-in PDF viewer for the Firefox web browser. At the time, web browsers depended upon separate reader applications or browser plug-ins to view PDFs.

Unfortunately, this created several security risks. External plug-ins can contain malicious code or gather data that could [endanger privacy](https://www.kaspersky.com/blog/browser-extensions-security/20886/). Downloading PDFs for local viewing is also potentially hazardous because it means the file must be removed from a secure application environment.

PDF.js uses Asynchronous JavaScript and XML (Ajax) to render PDFs as an [HTML5](https://www.accusoft.com/resources/blog/html5-document-viewer/) <canvas> element directly within a web application. Since it uses JavaScript for rendering, PDF.js is compatible with all modern browsers and doesn’t require any additional plug-ins.

In addition to being integrated into Firefox, the software was also made available as open-source code. This made it possible for independent developers to expand upon the core capabilities of PDF.js in the years since its release.

Should You Build or Buy a PDF.js Viewer?

The open-source availability of the PDF.js library makes it an attractive solution for software teams looking to add native viewing functionality to their applications. As with many open-source frameworks, however, developers may quickly run up against a few complications when building out a viewing solution from scratch.

Out-of-the-box, PDF.js consists of three basic layers:

* **Core Layer:** The heart of the JavaScript PDF library, this layer parses and interprets binary instructions from the file itself.
* **Display Layer:** This interface handles the actual rendering of the PDF into a <canvas> element.
* **Viewer Layer:** The primary viewing interface that allows users to view and interact with the document.

While the core and display layers can handle most documents, PDF.js doesn’t support the full [PDF specification](https://www.accusoft.com/resources/blog/a-brief-history-of-pdf-file-viewers/) and sometimes struggles with rendering lengthy, complex, or image-heavy files. Overall performance is often on the slow side, and the way text is rendered makes text search somewhat unreliable.

More importantly, PDF.js lacks out-of-the-box mobile support. The included viewer doesn’t provide essential mobile UI features like pinch-to-zoom. It also doesn’t respond dynamically to mobile screens to ensure that menus and tools remain usable on all devices.

Any developer looking to add PDF viewing capabilities to their web applications using PDF.js will need to solve these core issues. While features like responsive, mobile-friendly viewing may have been less important when PDF.js first released in 2011, they are considered essential by most users today. Unfortunately, building out these capabilities takes time and resources, which is something that few development teams have in abundance.

Integrating a ready-made viewer that combines the solid foundation of PDF.js with the innovative features users expect allows developers to quickly meet their project needs without pulling attention away from key aspects of their application.

PDF.js and Accusoft PDF Viewer

When the time came to develop a [lightweight PDF viewer](https://www.accusoft.com/products/pdf-collection/accusoft-pdf-viewer/) that could be deployed quickly as a 100% client-side solution, the Accusoft engineering team elected to use PDF.js as a starting point. With a decade of open-source development already behind it, the PDF.js library offered extensive resources that provided us with a great deal of flexibility as we designed a new viewing interface and worked to improve performance in terms of speed and file support.

We set out with three primary goals as we designed our new JavaScript viewer:

**Better Performance**

Existing PDF.js viewers tend to struggle with large and complex documents. They also don’t offer good text search support. We drew upon years of experience with image processing to enhance rendering quality, which included adding support for high DPI screen displays. These improvements allow the viewer to render text and vector based elements more sharply. We also implemented a number of enhancements to boost search result speed.

**Responsive Mobile Support**

Out-of-the-box mobile functionality is no longer a “nice-to-have” feature for a PDF viewer. That’s why we focused on building a user interface that not only provides key touch controls like pinch-to-zoom and swipe gestures, but also seamlessly adapts to various screen sizes. Since PDF.js doesn’t offer responsive viewing capabilities, we worked hard to design a viewer that would offer the same level of flexibility regardless of the device being used.

**Faster Integration**

While PDF.js has always been a lightweight PDF solution, the challenge comes with building additional features to support and expand its functionality. Building out a new viewing layer and implementing various performance improvements could not come at the expense of significantly increasing the JavaScript PDF library’s footprint. With no external dependencies, plug-ins, or complicated server configurations, it’s possible to integrate Accusoft PDF Viewer quickly without impacting overall application performance.

Experience the Evolution of PDF.js

While PDF.js has long served as an adequate open-source PDF viewing solution for web applications, today’s average user simply requires more functionality than PDF.js can provide on its own. For developers who lack the time, resources, or expertise necessary to build those additional features, PDF.js-based integrations like [Accusoft PDF Viewer](https://www.accusoft.com/resources/fact-sheets/accusoft-pdf-viewer/) can provide everything they need to add powerful PDF capabilities to their software solutions.

As a free-to-use integration, Accusoft PDF Viewer can be incorporated into an application by adding just a few lines of code. The JavaScript PDF library runs completely within your application environment, allowing you to retain full control over files and avoiding any burdensome or risky external dependencies. [Download Accusoft PDF Viewer](https://www.accusoft.com/products/pdf-collection/accusoft-pdf-viewer/#get-started) today to see the evolution of PDF.js first-hand!