Instant In-app Previews of Spreadsheet Files

Spreadsheet Viewer

Table of **Contents**
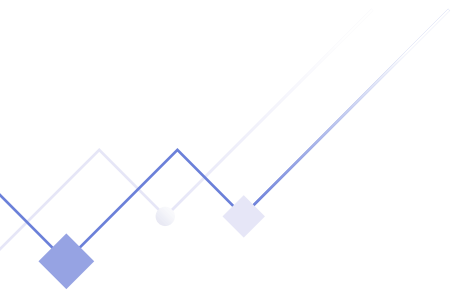
The Problem
With Excel Files

"Excel is everywhere you look in the business world—especially in areas where people are adding up numbers a lot, like marketing, business development, sales, and, yes, finance."

— James Kwak, "The Importance of Excel"

Microsoft Excel is the gold standard of productivity software, with more than half of Microsoft Office's one billion users worldwide using Excel [1]. In fact, it's so common in business that 63% of US companies rely on it in budgeting and accounting [2], and 81% use it in conjunction with their CPM solutions [3].

But when it comes to productivity, viewing Excel files gets complicated. There are two common ways to display the contents of Excel files, and both have their shortcomings:

1. Using external apps to view spreadsheet files.

This comes with considerable security risk for data owners. Files are first copied to the user's system or a cloud platform, and then opened using third-party spreadsheet apps like the desktop or online versions of Microsoft Excel, Google Sheets, and others. Plus whenever your users want to view a spreadsheet, they're taken out of your app.

2. Using an in-house component to view spreadsheet files.

With no out-of-the-box solution for this, developers need to either adapt open source libraries, or retrofit other tools – such as PDF viewers – to display Excel files. Apart from high development cost, the results for end-users are rarely an accurate representation of the original Microsoft Excel file.

1 Bryan, J. (2016). spreadsheets. Retrieved 9 May 2020, from https://speakerdeck.com/jennybc/spreadsheets?slide=7.

2 Robert Half. „Benchmarking Accounting & Finance Functions: 2019" Retrieved 14 January 2020.

3 MarketersMEDIA. „Ottawa Excel Consulting Expert Pivot Table Planner" Retrieved 14 January 2020.

An Interrupted UX
Is a Bad UX

As software becomes more mature and dynamic, the user experience emerges as the key differentiator, with a bad UX able to break an otherwise good digital product. Research shows that interruptions in how users engage with software disrupt both what they do and how they feel about it [4].
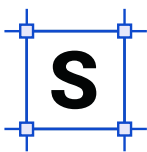
S hort-term memory in humans is only about 15-30 seconds. If a user has to first download a file, then search for it on the hard drive before opening it in a dedicated application, the cognitive load to resume what they were doing becomes too much.

Which is why product owners look for integrated solutions that enable users to perform all common actions within a single application.

To meet that objective, many apps have a quick file preview option that allows users to glance at the contents of a file without the annoyance of first opening it in an external app.
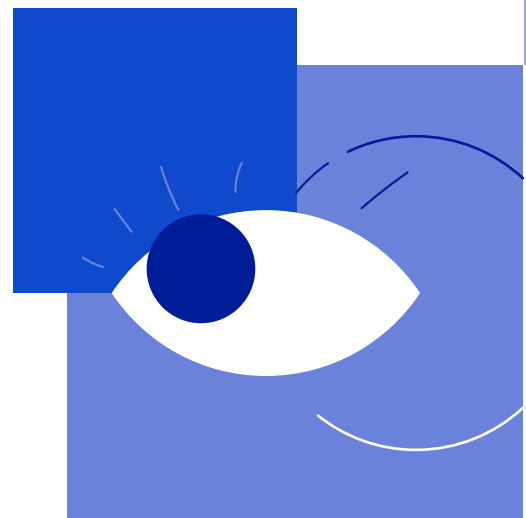
4 University of Nevada, Reno. "The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface". Retrieved 14 January 2020.

Discover a
Disruption-free Solution
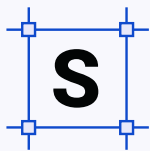
Spreadsheet Viewer addresses these problems by allowing users to view Excel files in read-only mode in any desktop or mobile browser. As a software vendor, you become independent of third-party cloud or software providers to properly view the contents of any spreadsheet file.

YOUR WEB APP

XLSX DOCUMENT

Spreadsheet Viewer is a JavaScript app that software developers and vendors can integrate into their own web app, making it possible for the contents of any XLSX file to be viewed directly in their app.

It can be integrated to enable viewing files in a modal window or inline with any other components of your app. Spreadsheet Viewer easily handles essential features such as cell formatting, charts, pivot tables, multiple worksheets, and more [5].

[5] Some features are work in progress. Contact us for details about the the product roadmap and the current feature set.

Key **Features**

- Accurate rendering of XLSX files
- Great user experience
- Tight data security
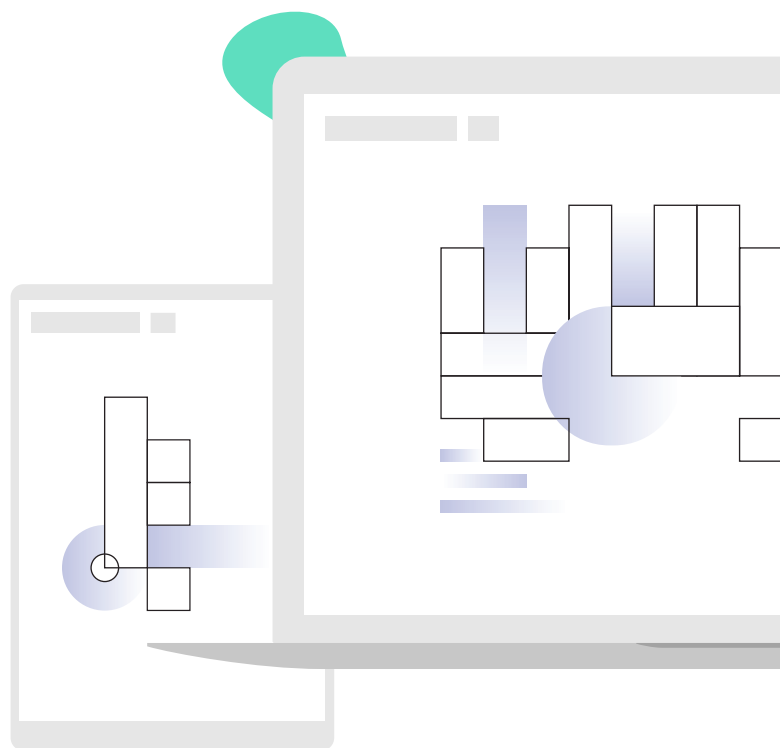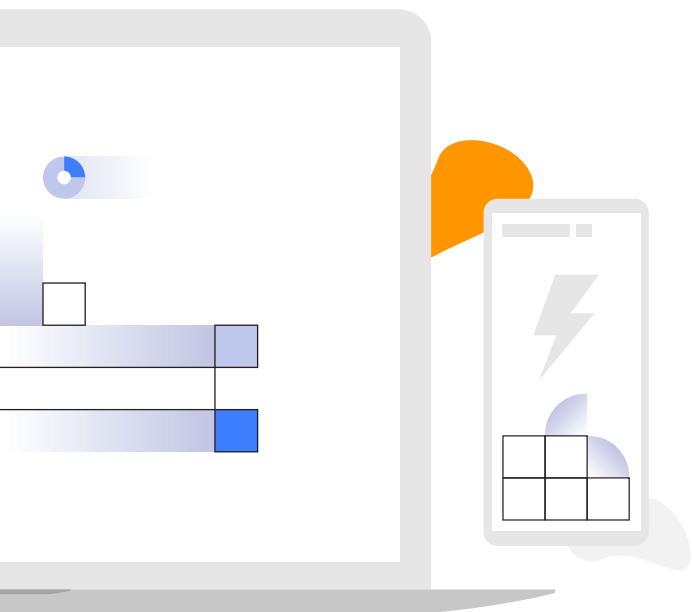
Benefits **for your Businesses**

- A turnkey solution
- Customizable
- Increases user engagement with your app
- Flexible license
- Continuously improved
- Dedicated commercial support
- Extra onboarding, training, and consulting available

Developer-friendly

- Comprehensive documentation including actionable guides and code samples
- Universal JS API that integrates with Angular, Vue, React, or any other framework
- Customizable look and feel
- Can be invoked in any display mode, such as a modal, popup, full-screen, or inline
- Cross-platform, including mobile
- Works offline, directly in-browser

Benefits
for your Users

- Quick preview on any device
- Always ready to use, no additional software needed
- Familiar Excel-like experience, with tab navigation and keyboard shortcuts
- Easy clipboard copying to other apps
- Loading indicator while files are transferred and prepared for rendering
- No clutter in the Downloads folder
- Previewing before downloading allows the user to ensure they download the correct file

Use Cases

Inbox, Messaging, and Online Meeting Apps

Many inbox and messaging app providers struggle with properly displaying advanced attachment formats, including Excel files. With Spreadsheet Viewer, you can add this critical feature, improving the overall user experience. Users can view the exchanged files without leaving the inbox or chat window.

Enterprise Software

Whether you are a vendor of an integrated ERP suite or a best-of-breed B2B application, your users will commonly resort to Excel files in parts of their workflows or reports. With Spreadsheet Viewer, previewing them is not only at their fingertips, but also completely secure.

File Hosting Services

Spreadsheet Viewer improves workflow, giving users instant access to a functional file preview of their spreadsheet files without saving them on their hard drive.

Product and Project Management Software

Tickets in project management software can include attachments like roadmaps, resource plans, client requirements, or translation keys. The viewer will allow end users to preview the files instantly without having to download them.
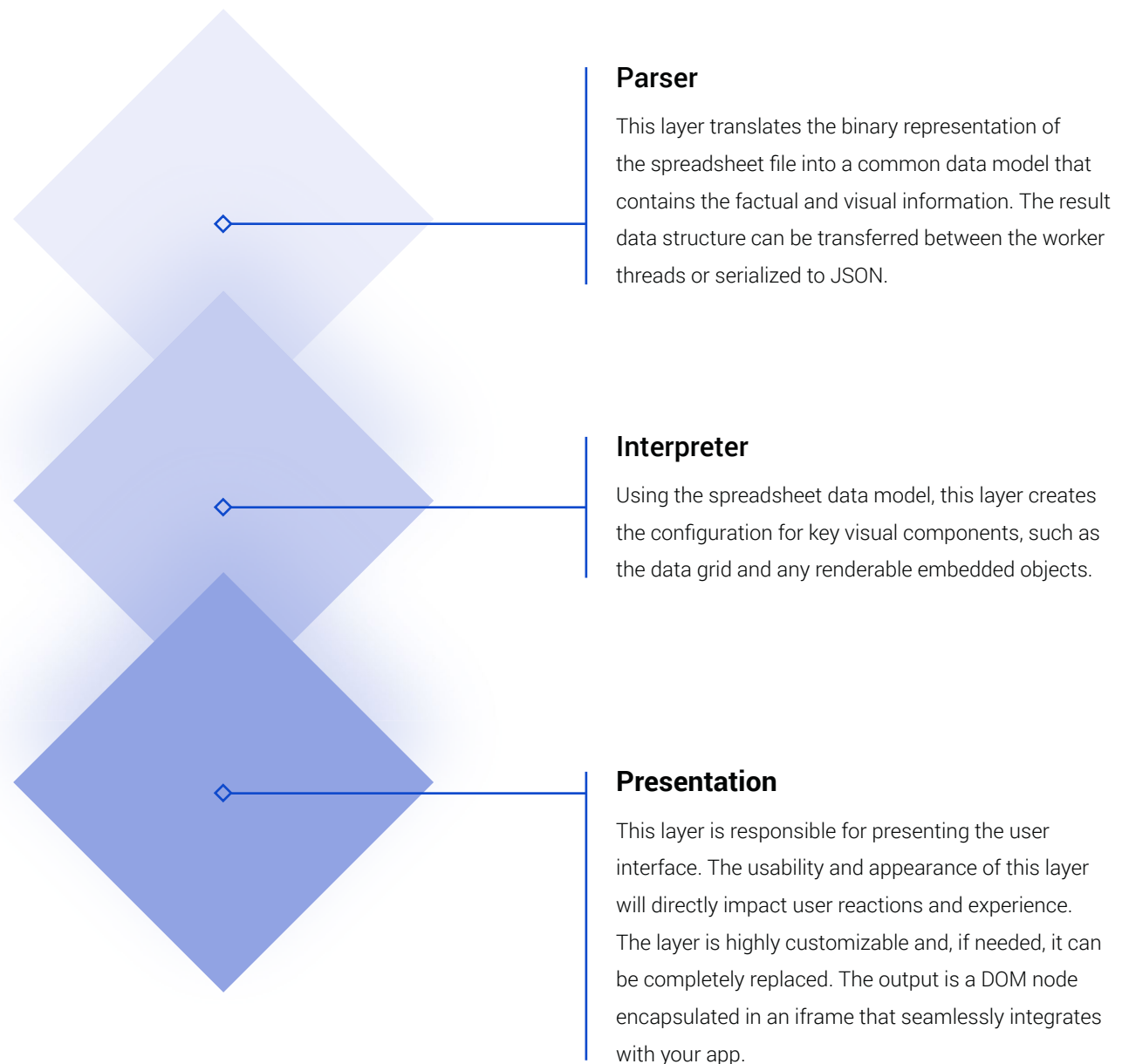
Technology

Product Architecture

The product's modular architecture consists of three layers, preconfigured to run as a turnkey solution that integrates with any app. The layers are independent, allowing for easier customization of the product. The role of each layer is presented on the diagram.

Parser

This layer translates the binary representation of the spreadsheet file into a common data model that contains the factual and visual information. The result data structure can be transferred between the worker threads or serialized to JSON.

Interpreter

Using the spreadsheet data model, this layer creates the configuration for key visual components, such as the data grid and any renderable embedded objects.

Presentation

This layer is responsible for presenting the user interface. The usability and appearance of this layer will directly impact user reactions and experience. The layer is highly customizable and, if needed, it can be completely replaced. The output is a DOM node encapsulated in an iframe that seamlessly integrates with your app.

Platform Requirements

- OS: Any mobile or desktop operating system capable of running a modern web browser or web view. Android, macOS, Windows, iOS, and Linux are officially supported.
- Web browsers: Any modern web browser. Chrome, Firefox, Opera, Safari, Internet Explorer 11, and Edge are officially supported.

Security

- Works entirely on the client side.
- The files are presented in a secure sandbox of the web browser, without macros, so they can't make any changes to the user's system
- For an additional layer of security, you can deploy Spreadsheet Viewer on a separate host, which will fully isolate it from the rest of your app thanks to the iframe cross-origin policy.
- Spreadsheet Viewer uses third-party code with clear licensing. Public security audits are available via npm audit and Snyk.

Performance

- Total library size: under 900 KB gzipped [6]
- The mean parsing time is just **14** milliseconds, and 90% of files are parsed in under **345** milliseconds (test conducted on a i7-8700k CPU). The chart below shows the parsing time of 1679 typical XLSX files in relation to their size.
- Fully cacheable. Once a preview is loaded, all of the resources needed for future rendering of the spreadsheet previews can be stored by the browser. Caching is controlled by HTTP headers sent by the static file server.

Parsing time of 1679 typical XLSX files of various size
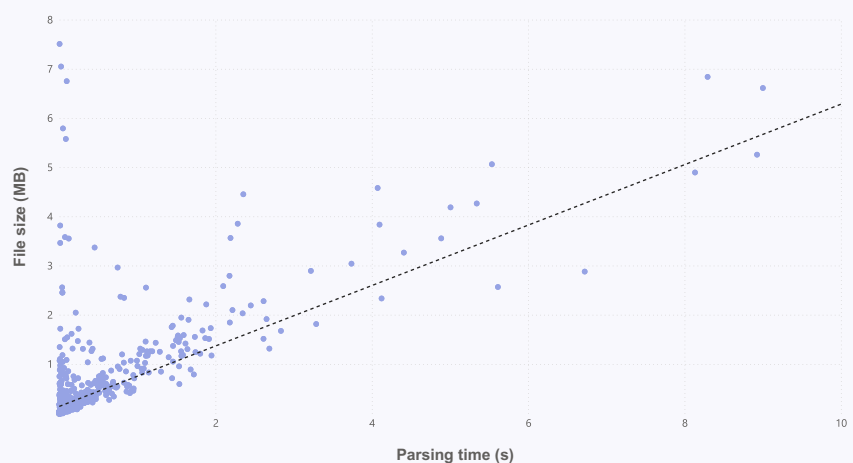
[6] Excluding extra assets such as charts and web fonts.

Integration

The quickest way to enable the viewer in your software is by adding a short code snippet to the end of the <body> section of the main HTML document.

The easy integration with your web app is enabled by a JavaScript **API** that is compatible with any framework. Using a simple, asynchronous interface, the library creates an encapsulated app container (iframe) and handles the loading of the application **assets** from a web server as well as the spreadsheet files from a given **URL** or an **ArrayBuffer**. This allows rendering any file stored in a database or cloud, or generated on demand, etc.

We describe the configuration of the presentation layer in detail in the product documentation.

The source code of the presentation layer is available upon request, which means you and your team can introduce changes or create your own version, including incorporating your corporate design system.

The viewer doesn't collect analytical data, but user interactions such as loading of files, changing active sheets, or even copying to clipboard events can be handled in your app to analyze stats such as clicks, etc.

We encourage interested individuals to contact us to learn about the possibilities of server-side parsing or rendering.

Product Development Timeline

- First public release – available now
- More formats and features – **coming up in 2021**

Get in Touch

If Spreadsheet Viewer looks like a match for your software, let's talk about how you could use it for your business during a screen-sharing call. We'll also explain the relevant technical details and pricing.

To schedule a call, send an email to **Marcin Warpechowski** marcin@handsontable.com

About Us

Handsoncode is a software development company founded in 2015. We're the authors of Handsontable, one of the most popular JavaScript data grid libraries in the world. We have more than 1,500 customers worldwide, including many from the Fortune 500 list.

ppnt gdynia

Our team consists of 20+ dedicated specialists who understand and embrace creating developer tools.

We provide consulting services, on-demand development, and stellar multi-channel customer support.
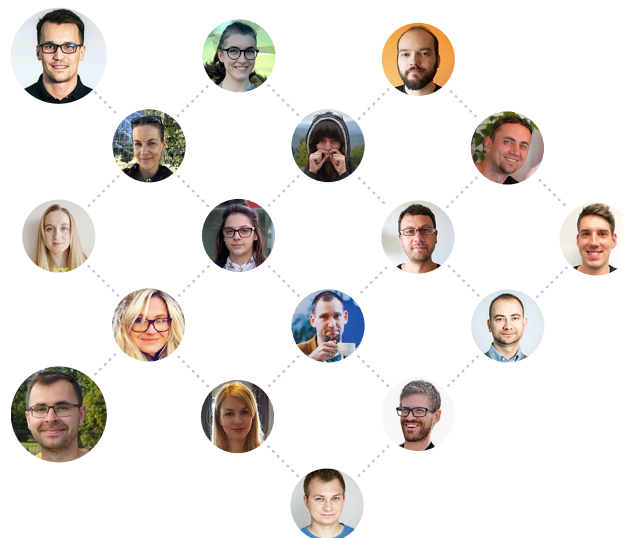
Disclaimer

- Handsontable Spreadsheet Viewer is a trademark of Handsoncode sp. z o.o.
- This document is not a binding offer. We're sending you marketing information with an option to get in touch with us. The listed product features may change.
- All trademarks used in the document belong to their rightful owners.