

Introducing LaTeX Base

Gareth Aye

July 4, 2016

[LaTeX Base](#) is a web-based \LaTeX editor that provides many useful features such as

- live, compile-as-you-type document preview,
- one-click document publishing and sharing,
- offline mode,
- integrations with file storage services like Google Drive and Dropbox,
- syntax highlighting,
- and familiar keyboard shortcuts for Vim and Emacs users.

This article focuses on the technical side of the interactions between LaTeX Base and \LaTeX . However, we're eager to hear from users about their experiences using the service; please feel free to reach out to us at team@latexbase.com with suggestions or if you'd like to beta test experimental, new features.

1 Offline mode

One unique aspect of LaTeX Base amongst web applications is that it can be used with or without an Internet connection! This is possible thanks to service workers: a recent development in the web platform that allows applications to intercept and cache network requests. That means that you can write your papers on a plane, in the park, or anywhere in between. What's more is that, while LaTeX Base isn't the first web-based \LaTeX editor, it is the only one with this capability.

If you're a programmer as well as a \LaTeX enthusiast, you may suspect that there's even more to the story. Service workers allow us to cache the editor, but \LaTeX doesn't run in the browser (it's a C program). How can LaTeX Base compile documents without an Internet connection if it's limited to running code in JavaScript?

2 Enter Emscripten

The answer lies in a fascinating tool that came out of Mozilla's research group a few years ago called [Emscripten](#). Emscripten compiles LLVM bytecode to JavaScript. Many codebases such as Unreal Engine, Bullet Physics, and the Lua programming language (which appears to be of some interest to the greater \LaTeX community) have been ported from C/C++ to JavaScript using Emscripten. In building LaTeX Base, `pdf \LaTeX` was compiled to LLVM bytecode using `clang` and from LLVM to JavaScript using Emscripten. Whereas similar services send users' documents to servers with installed \LaTeX compilers and packages, LaTeX Base sends the compiler and packages to the browser. In addition to making offline mode possible, compiling in the browser also allows LaTeX Base to compile documents quickly and often – so much so that we can offer a real-time preview instead of forcing the user to compile manually.

3 Packages

Our design goal when considering the issue of packages was to make a large number of packages (eventually anything hosted on the CTAN registry) available to users while only ever downloading the packages needed to compile their documents. What we came up with is lazy package loading. Every time you include a new package in a document with LaTeX Base, you'll download it from our servers. When you use that package in the future it'll be cached in your browser. For this reason, using packages that a user hasn't previously used while offline will not work. You can also expect compiling to take slightly longer the very first time you use a package. For the time being, we only support a small number (around 25) of the most commonly used packages, but our roadmap includes extending support to arbitrary hosted packages.

4 Images

The only way that our implementation of \LaTeX differs from a standard compiler is in how we handle external files (like images). We don't currently give users direct access to the virtual Emscripten filesystem that \LaTeX sees when it's running on <https://latexbase.com>. Instead, we support calling `\includegraphics` with an image url that we'll fetch and preload in Emscripten's virtual filesystem. When you download your documents, we automatically convert these url identifiers to simple file names and bundle the downloaded images so that no changes are necessary to compile documents elsewhere. Our roadmap also includes allowing users to upload images and other local resources rather than supplying urls.

5 Conclusion

Web applications are great. They allow users to use software without permanently installing it. They're very limited (compared to native applications) in the abilities and permissions they're granted by default, so they are preferable from a practical security perspective. They're built on open standards, so they run anywhere and don't need to be sanctioned by any organization or app store. Using cutting edge tools, LaTeX Base is able to offer many advanced capabilities right in the browser. In this author's (absolutely biased) opinion, it's on its way to becoming the best way to write \LaTeX docs.

In closing I want to recognize Mozilla not only for their tremendous standards work that's made the web the wonderful thing it is today, but also for their work on components that made LaTeX Base possible including Ace, Emscripten, PDF.js, and localforage.

6 About the author

Gareth is a New York native living in Portland, OR with his wife Alison and toddler Albee. He received a BA in Computer Science with a Math minor from Middlebury College and worked in software development, most recently as an engineering lead at Mozilla before building LaTeX Base. In his free time, he enjoys playing jazz piano and chess.