

Minimal Computing with Progressive Web Apps

Chris Diaz <chris-diaz_at_northwestern_dot_edu>, Northwestern University  <https://orcid.org/0000-0002-5247-2718>

Abstract

Caravans of Gold is a multilingual digital exhibit that was built as both a static website and Progressive Web App (PWA). This case study describes the process of developing *Caravans of Gold*, situating PWAs as minimal computing technologies. PWAs are websites with mobile app features that specifically help disseminate digital content to people who rely on mobile devices for access to the internet. *Caravans of Gold* was built using Wax, a Jekyll-based framework, and demonstrates the similarities between PWA requirements and minimal computing approaches.

INTRODUCTION

In 2019, the Block Museum of Art at Northwestern University presented *Caravans of Gold, Fragments in Time: Art, Culture, and Exchange across Medieval Saharan Africa*, an exhibition of collection items representing trade and cultural exchange across the Saharan desert from the 8th to 16th centuries [Berzock et al. 2019]. That spring, the lead curator of *Caravans of Gold* led a seminar in which the students were tasked with developing content for a digital version of the multilingual exhibit as a class project. The goals of the digital exhibit were twofold: (1) provide undergraduate students with an opportunity to make curatorial decisions in a real-world context and (2) develop a digital museum experience for people in Mali, Morocco, and Nigeria. Northwestern University Libraries partnered with the Block Museum to develop and maintain the exhibit website.^[1]

1

PROJECT PLANNING

More than half of all global internet traffic is generated from a mobile device. In African countries, mobile internet traffic accounts for two-thirds of all web page visits [Statista 2021]. In terms of network speeds, Morocco is about even with the global average; however, mobile data in Nigeria and Mali is downloaded at half the speed of the global average [Speedtest a] [Speedtest b] [GSMA Mobile Connectivity Index]. This was echoed anecdotally from our museum partners,^[2] who requested a mobile app, rather than a traditional website, for the digital exhibit.

2

Traditional mobile apps are developed for the operating system on which they are intended to be installed. Android is the most popular operating system in Mali, Morocco, and Nigeria; however, iOS is also used in Nigeria and Morocco, with between 7% and 19% of mobile market share respectively [Mobile Operating System Market Share Worldwide]. Unfortunately, the same native mobile app cannot run on both Android and iOS devices; thus, we would need to develop two mobile apps with identical features.

3

WHAT DO WE REALLY NEED?

Given this scope, it was useful to unpack the initial request to develop “a mobile app” for *Caravans of Gold*. Northwestern University Libraries does not develop mobile apps. We develop static websites for various forms of scholarly and educational materials and deposit archival copies in an institutional repository. Static websites help keep our costs down while giving us the flexibility to produce customizable, modern, secure, and content-rich websites for

4

students and scholars.

It is here that the central question of minimal computing — “What do we need?” — saved this project idea from being rejected outright [Gil and Ortega 2016]. These were the core requirements:

- Designed for mobile devices;
- Performant on low-bandwidth networks;
- Installable on mobile phones, leaving an app icon on the device;
- Usable while not connected to the internet; and
- Maintainable for at least 10 years.

It turns out that we did not need the complex capabilities of traditional mobile apps, such as user accounts, access to the device’s file system, control of the camera or microphone, or push notifications. What we needed was a static website with Progressive Web App (PWA) features.

Fortunately, the planning phase for *Caravans of Gold* happened to coincide with early news of Wax, “a minimal computing project for producing digital exhibitions focused on longevity, low costs, and flexibility” [Nyrop and Gil]. Wax is an open source framework for the Jekyll static-site generator that contains a website theme and a set of automations that can be used to make a static website from a folder of images and a spreadsheet of item metadata. Wax made it easy for us to get started, and its connection to minimal computing generated learning opportunities for the libraries, the museum, and the students.

As a class project, the students assumed the role of museum curators by selecting images, writing texts, and working with a librarian to make a digital exhibit app. During the quarter, we spent a class period discussing the basics of minimal computing and the process for using Wax to build the exhibit. At the end of the term, the students assembled a folder containing photographic images of museum exhibit items, a spreadsheet of metadata about the items, a text document with some background, and contextual information for the website pages.

PROGRESSIVE WEB APPS (PWAs)

PWAs provide the option for an app to be installed on a device for quick access and offline use. Unlike traditional mobile apps, PWAs can be written in standard HTML, CSS, and JavaScript. There is no formal specification, nor is there a canonical package, library, or framework. PWAs simply denote websites that meet the following criteria:

- The website uses responsive design to fit any screen size;
- The website is served over an HTTPS network;
- The website includes a service worker to intercept and respond to network requests; and
- The website includes a web app manifest file to allow devices to install it (MDN Contributors).

Wax produces a responsive website that fits any screen. The HTTPS requirement is a feature provided by the web hosting service (in our case, Amazon Web Services). The web app manifest and service worker files needed to be added to the website we built with Wax and the Jekyll Multiple Languages plugin.

A web app manifest is a JSON file that includes information related to how an application can be installed to a device [MDN Contributors]. A typical web app manifest includes the full name of the application; a short name of the application; a list of app icon paths in various sizes for mobile, tablet, and desktop screens; a background color; a display setting; and a URL for the app landing page. We provide three web app manifests, one for each language in the exhibit.

A service worker is a JavaScript file that acts as a network proxy to respond to HTTP requests, such as navigating from one web page to another [Gaunt 2019]. These files eliminate unnecessary network calls and enable the digital exhibit to be usable while the device is offline.^[3] We used a service worker to precache all HTML, CSS, and JavaScript files for the English, French, and Arabic versions of the website and cache only the images that are loaded to the screen.

PWAs are an example of minimal computing concepts put into practice. Alex Russell distinguishes between PWAs and other web applications by their adherence to a set of foundational principles:

13

- URLs and links as the core organizing system: if you can't link to it, it isn't part of the web;
- Markup and styling for accessibility, both to humans and search engines;
- UI richness and system capabilities provided as additions to a functional core; and
- Free to implement without permission or payment, which in practice means standards-based. [Russell 2015]

Put into practice, these principles resemble Jentery Sayers's definitions of "minimal" and "maximal" forms of computing: maximum access, maximum accessibility, minimal connectivity, and minimal vulnerabilities [Sayers 2016]. Native desktop and mobile apps, by contrast, betray these principles with proprietary packaging, installation requirements, and permissions-based distribution through app stores.

14

By side-stepping the app-store infrastructure, PWA evangelists seem to espouse similar values that we hear from the open access movement [Chan et al. 2002]. In both cases, there is a desire to emphasize the basic technologies of the web in order to facilitate the free distribution of information. While the popularity of PWAs is indeed driven by commercial interests, the use of services workers and web asset bundling tools for academic web publications can make the renewal, dissemination, and preservation of the scholarly record easier.

15

SPEED AND PERFORMANCE

Websites that are slow to render are more difficult and frustrating to use, so we made it a priority to ensure fast page load times for *Caravans of Gold* users in Mali, Morocco, and Nigeria. Performance testing measures the speed at which the content of the website is loaded and displayed on a variety of user devices and networks. This is a key requirement to the PWA development process. Our primary tool for measuring performance is Google's Lighthouse performance audits. Lighthouse provides evaluations in several categories: performance, accessibility, best practices, search engine optimization, and PWA. Lighthouse will simulate a desktop or mobile device and load the web page you select, assessing any one of those areas on a scale of 0-100. We made the following improvements as a result of the performance testing:

16

- Deploy the website files (HTML, CSS, JavaScript) with GZIP compression^[4] and
- Use tree shaking to analyze and remove unused CSS code^[5]

These tests and improvements provide some assurances that any visitors in Mali, Morocco, and Nigeria will have a good user experience while accessing the exhibit.

17

THE PROBLEM WITH IMAGES

Large images consume high amounts of network data and make websites slow to load. *Caravans of Gold* contains over two hundred images but we wanted to keep the exhibit as small as possible by using a minimal number of image files and smaller file sizes, to ensure quick load times on any device. Meeting this goal prevented us from using some of the image processing features of Wax (`wax:derivatives`).

18

The `wax:derivatives` automation script is intended to process images for use on the web. There are two options for running the script called "simple" and "IIIF." The simple option uses an image to create two derivatives: (1) a small, thumbnail version and (2) a full, resized version. The IIIF option, named after the International Image Interoperability Framework (IIIF), processes the images according a technical specification for making high-resolution images reusable by scholars and institutions. IIIF is typically used with image repository systems. It works by serving an image as many small image "tiles" that, all together, comprise the full image, allowing users to zoom, annotate, and manipulate the images for a variety of scholarly purposes. Both of these options generate more image files of larger file sizes than we need. One of the primary goals of PWAs is to lessen the hardware and network requirements for users of websites, and large images would require *Caravans of Gold* users to have more device storage and network data access than is

19

necessary.

Instead, we resorted to compressing and resizing each image individually.^[6] We would not recommend this approach if it can be avoided. The original *Caravans of Gold* images varied in size, quality, and format. We needed to work with each image individually in order to crop, resize and reformat it in preparation for use with the project. It is generally a best practice to serve images that are appropriately sized to fit the screens upon which they will display. For the majority of our images, we created one size for desktop screens (between 600px and 800px in width) and one size for mobile screens (300px in width). We used HTML markup to help browsers select the appropriate image based on the width of the screen.^[7]

20

TRANSLATIONS

Because *Caravans of Gold* features items that are loaned by museums in Mali, Morocco, and Nigeria, the Block Museum commissioned the translation of the English-language content into French and Arabic, the official languages of Mali and Morocco respectively. The goal was to enable all of the international lending institutions to promote the exhibit to their patrons, rather than exclusively publishing the exhibit in English.

21

Wax does not support multi-language websites. Wax provides an automated workflow for generating website content from a spreadsheet, but this only works on monolingual websites. Working with translations required us to use external tools to help generate the website in multiple languages. To handle the Arabic and French translations, we needed the Jekyll Multiple Languages plugin [Kurtson], which creates standalone websites for each language.^[8] The plugin stores all of the translations in YAML files, which are plain text files containing key-value pairs (for example, “title: Caravans of Gold”).

22

You can think of the YAML file as a database of definitions, where each word or phrase (the key) has a definition in either English, French, or Arabic (the value). For example, the English definition of the title of the website is represented as “title: Caravans of Gold” in the YAML file. Every word and sentence in the English, French, and Arabic versions of the website needed to be represented as an entry in a YAML file. This meant that we had to copy and paste text from Microsoft Word documents (.docx files) that were provided by the translators.

23

While the plugin allows us to build a website in three languages from a single codebase, we learned that it did not support right-to-left languages, like Arabic, without customizing the layouts. We had to significantly alter the HTML layouts in order to render an adequate right-to-left reading experience for the Arabic version of the exhibit. We did this in two ways: adding a `dir="rtl"` attribute to HTML elements and using CSS rules to right-justify the text [Ishida 2016].

24

GUIDED TOURS

We designed *Caravans of Gold* to provide three methods of navigation: browsing, searching, and touring. The Wax theme organizes the exhibit items for browsing and searching by providing a gallery view of the collection as well as a search interface, but we also wanted to simulate a guided tour experience for visitors interested in reading a narrative behind the works, history, and locations featured throughout.

25

The exhibit is essentially organized into thirteen tours representing locations along a major Saharan trade route and points in time during the medieval period. Each tour features several key works and important context about the work’s history, relevance, and excavation. When a user clicks into one of the tours, they are launched into a full-screen slide deck presentation.^[9] Each tour is a single HTML file controlled by JavaScript to manage advancing through the slides. This was the final content piece we added to the exhibit.

26

CONCLUSION

The average internet user in the Global North has a faster network connection and more powerful internet device than the average internet user in the Global South. Static websites and PWAs provide technical solutions for addressing these global inequities in user experience by minimizing load times, file sizes, and network requests. Static website

27

frameworks, like Wax, can be extended to include PWA features in order to improve the access and usability of digital content to people in the Global South. *Caravans of Gold* combines minimal computing with PWA technologies to successfully produce a multilingual digital exhibit app optimized for people around the world.

Notes

- [1] The final exhibit website is available at <https://caravans.library.northwestern.edu/>. The source code for the final website is available at <https://github.com/nulib/caravans-wax>.
- [2] The list of lending institutions is available at <https://caravans.library.northwestern.edu/about/>.
- [3] We used Google's Workbox software to create our service worker.
- [4] We used the `s3_website` Ruby gem to manage deployments to our Amazon Web Services hosting.
- [5] We used PurgeCSS, but tools like Webpack and Parcel are also very popular.
- [6] After we correctly sized and cropped images, we used the public TinyJPG and TinyPNG web services for the compression.
- [7] The `srcset` attribute for the `` element provides the browser with both large and small options for the same image [Coyier 2014].
- [8] For example, the French-language exhibit is written to a directory called "fr" and deployed to <https://caravans.library.northwestern.edu/fr/>.
- [9] We used the RevealJS slide deck presentation framework for HTML to create custom Jekyll Layouts to supplement the templates Wax provides.

Works Cited

- Berzock et al. 2019** Berzock, K.B., Estrela, S., Sympson, M.G., Diaz, C., Andrey, E., Biggs, L., Considine, M.C., Cook, B.R., Detweiler, Z., Heath, B., Liou, N., Milaz, M., Simon, J., Qian, C., & Zhang, E. *Caravans of Gold, Fragments in Time: Art, Culture, and Exchange across Medieval Saharan Africa*, 2019. <https://caravans.library.northwestern.edu/>.
- Chan et al. 2002** Chan, L., Cuplinskas, D., Eisen, M., Friend, F., Genova, Y., Guedon, J.-C., Hagemann, M., Harnad, S., Johnson, R., Kupryte, R., La Manna, M., Rev, I., Segbert, M., de Souza, S., Suber, P., Velterop, J. "Read the Budapest Open Access Initiative," 2002. <https://www.budapestopenaccessinitiative.org/read>.
- Coyier 2014** Coyier, C. "Responsive Images: If you're just changing resolutions, use srcset." *CSS-Tricks*, 2014. <https://css-tricks.com/responsive-images-youre-just-changing-resolutions-use-srcset/>.
- GSMA Mobile Connectivity Index** *GSMA Mobile Connectivity Index*. Accessed June 14, 2020. <https://www.mobileconnectivityindex.com/>.
- Gaunt 2019** Gaunt, M. "Service Workers: An Introduction." *Google*. Accessed April 10, 2022. <https://developers.google.com/web/fundamentals/primers/service-workers>.
- Gil and Ortega 2016** Gil, A., Ortega, E. "Multilingual Practices and Minimal Computing." In *Doing Digital Humanities: Practice, Training, Research*, edited by C. Crompton, R.J. Lane, and R. Siemens, 22-34. NY, Routledge (2016).
- Ishida 2016** Ishida, R. "Structural Markup and Right-to-Left Text in HTML." *W3C*, 2016. Accessed June 23, 2020. <https://www.w3.org/International/questions/qa-html-dir>.
- Kurtson** Kurtson, M. Jekyll Multiple Languages Plugin. Accessed September 22, 2021. <https://github.com/kurtsson/jekyll-multiple-languages-plugin/>.
- MDN Contributors** MDN contributors. "Web App Manifests." MDN Web Docs. Accessed June 24, 2020. <https://developer.mozilla.org/en-US/docs/Web/Manifest>.
- Mobile Operating System Market Share Worldwide** Mobile Operating System Market Share Worldwide. *StatCounter Global Stats*. Accessed June 14, 2020. <https://gs.statcounter.com/os-market-share/mobile/worldwide>.
- Nyrop and Gil** Nyrop, Mari. and Alex Gil. Wax. Accessed September 22, 2021. <https://github.com/minicomp/wax>.
- Russell 2015** Russell, A. "Progressive Web Apps: Escaping Tabs Without Losing Our Soul." *Infrequently Noted*, 2015. <https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>.

Sayers 2016 Sayers, J. "Minimal Definitions (tl;dr version)." *Minimal Computing Working Group*, 2016. <https://go-dh.github.io/mincomp/thoughts/2016/10/03/tldr/>.

Speedtest a "Morocco's Mobile and Broadband Internet Speeds." *Speedtest Global Index*. Accessed June 14, 2020. <https://www.speedtest.net/global-index/morocco#mobile>.

Speedtest b "Nigeria's Mobile and Broadband Internet Speeds." *Speedtest Global Index*. Accessed June 14, 2020. <https://www.speedtest.net/global-index/nigeria#mobile>.

Statista 2021 Internet Usage Worldwide. *Statista*. Accessed June 14, 2020. <http://www.statista.com/study/12322/global-internet-usage-statista-dossier/>.



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.