Here are some things I learned along the way that might be useful in creating something similar to Blocks at Lexington High School. If you have any questions at all, I'd love to help: just send me an email (ben@elk.sh).

Start small

First, some personal advice. **Start small.** When I first started sending around Blocks in the summer of 2019, all you could do was put in your classes and see today's schedule. No calendar, no weekly view, and you couldn't even see what classes you had tomorrow.

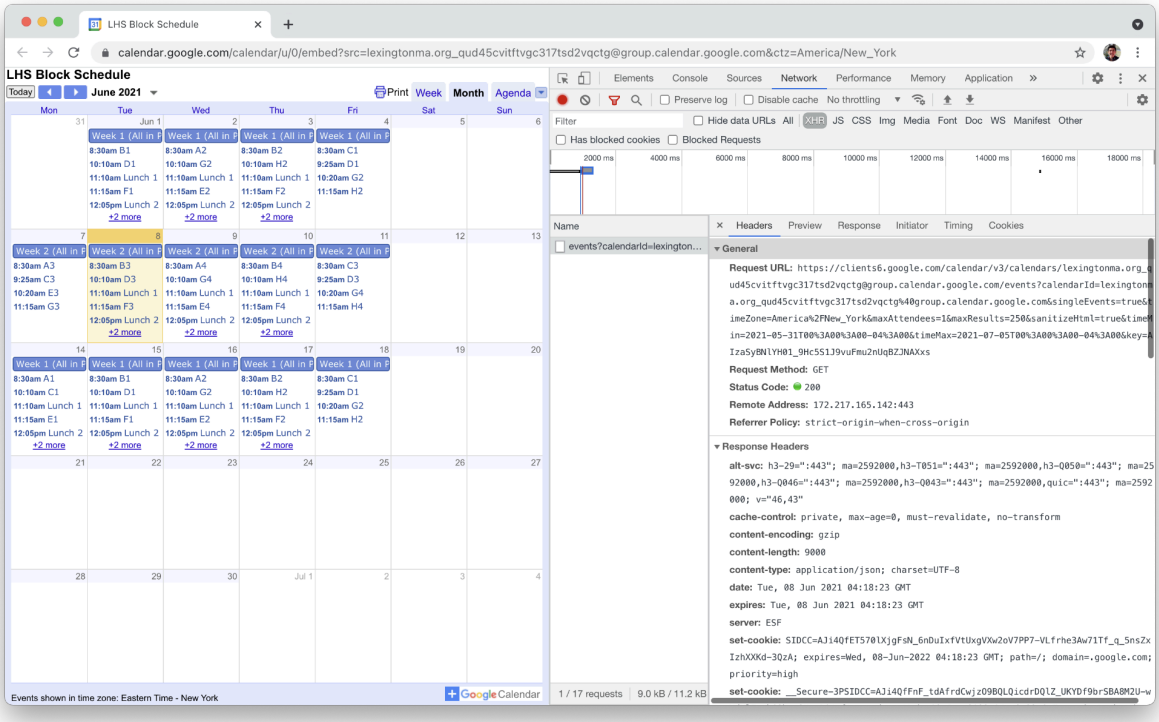
These things all came later. The important thing for me was to get it out there and have people get excited about it. If I hadn't told my friends about it and gotten 8 people to download the app while I was at dinner, I probably would've lost motivation.

There's a lot of features in Blocks now, and it might be intimidating to think that that's what people expect out of something you build. But I urge you to build a tiny version of it, get it out there, and let people get excited about it. Don't overwhelm yourself with building out scratchpads and sticky notes and custom theme colors and ways to check your grades. Start small, and let people's excitement and the ideas that they give you carry you forward.

How to use LHS' Google Calendar

LHS has a really robust Google Calendar that they update regularly: bit.ly/LHSBlock. Building on top of this will save you from having to update your copy of the schedule for every half day, unexpected schedule change, etc. For Blocks, I had a repeating cron job that downloaded this once a day at midnight, and then cached it for two days (make sure to cache it so you're not re-downloading the entire calendar each time someone uses the app).

I didn't want to mess with the Google Calendar API, so I just opened Chrome dev tools on bit.ly/LHSBlock and copied the URL for the request they make. Make sure the URL sets the timezone to America/New_York — that'll save you a lot of headache with daylight savings time.



I filter out all-day events, and replace the letters (e.g. “E4”) with the user’s inputted class for that block.

Class creation UI

In version 1 of Blocks, I had users create classes first, and then assign them to blocks (“A1”, “A2”, etc). This leads to a lot of duplication (users usually have to add the same class to A1, A2, A3, and A4).

In version 2, I had people create classes *and* mark which blocks they meet in at the same time. I like this approach because of its simplicity and speed, and you might want to consider adopting it as well. The only downside is that two different classes could be marked as the same block — you just have to keep this in mind, and pick one of the two to display in this edgecase.

Define a single source of truth of “now” time

It’s really useful if your entire app relies on one variable to get the current date and time (instead of re-getting the current timestamp every time). This way, you can “time travel”

and check what the entire app looks like on a future date just by changing that single variable.

Keep track of last day of school

You'll probably find this on one of the holiday calendars that LHS puts out. It's useful to know when to end the in-app calendar. Also, keep comments in your code about when you bump this up because of snow days. The school will not tell you when the new last day of school is, and you will forget whether or not you bumped it up for past snow days.

Standalone mode

Standalone mode is when the website is added to the home screen of an iOS or Android device. I didn't create native apps, because web apps are easier to update without going through both app stores, and only require one codebase.

You can detect whether the web app is added to a user's home screen in JavaScript like this:

```
if(window.navigator.standalone) return true
if(window.matchMedia('(display-mode: standalone)').matches) return true
return false
```

The first line checks for iOS, and the second for Android.

Blocks gives you different functionality on the homepage depending on what mode you're running in:

- Desktop: you can only log in, not register.
- Mobile but in-browser: you only get instructions for adding the app to the home screen (see instructions below).
- Mobile and standalone: you can both log in and register.

I did this to force every user to have Blocks on their home screen, so they wouldn't forget the URL.

Creating a progressive web app

This is the viewport meta tag that I used, which doesn't allow users to zoom in when the web app is added to their home screen:

```
<meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no, viewport-fit=cover, user-scalable=no">
```

I added and linked a web manifest.
Apple has specific meta tags that you should add:

```
<meta name="apple-mobile-web-app-capable" content="yes" />
<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent"
/>
<meta name="apple-mobile-web-app-title" content="Blocks" />
<link rel="apple-touch-icon" href="/images/icon.png" />
```

I also defined splash screen meta tags. The splash screen is the image that shows up while the web app is loading (for Blocks, it's the backpack emoji on a white background). There's a ton of different screen sizes you have to account for, but this splash screen generator does it for you.

Instructions for how to add to home screen

I show different instructions for how to add the web app to a user's home screen depending on if it's an iOS or Android device (and only if it's a mobile device).

```
const isIos = () => (/iPhone|iPad/i).test(navigator.userAgent),
const isAndroid = () => (/Android/i).test(navigator.userAgent),
const isMobile = () => obj.isIos() || obj.isAndroid()
```

Here are the instructions I use:

- If iOS and standalone: open in Safari, tap the "share" icon button and then scroll down to "Add to Home Screen".
    - Note: as far as I know, it's a "three dot" icon instead of a "share" icon on iOS 15.
- If Android and standalone: tap the "3 dot" menu at the top right in Chrome, then "Add to Home screen".

Scraping email addresses from Google Contacts

I only allowed users to sign up if they were in my directory of @lexingtonma.org email addresses. I got that directory by scraping Google Contacts for LHS. This was super helpful, because it didn't let anyone sign up with a mistyped email address or a personal email address, and I already had verified names for every account.

However, this did mean that I had to add a couple new students to the directory mid-school year so they could sign up (I customized the "you can't sign up" error message to tell them to contact me).

Make sure to filter out test email addresses, since LHS has some test users that show up in Google Contacts.

Naming

I'd appreciate it if you didn't use "Blocks" in the name of your app, to avoid confusion. Names that rhyme with Blocks or are a play on words are fine, though.

--

Once again, please reach out (ben@elk.sh) if you'd like to take the initiative to build something similar to Blocks and would like any help or questions answered. I also really want to hear about any app you build for LHS, so please shoot me an email if you do (seriously, it's not weird, please do it, it'll make my day).

Thanks!

— Ben Borgers