How I Doubled my Salary in Five Months and Got an Amazing Job

medium.freecodecamp.org/how-i-doubled-my-salary-in-5-months-and-got-an-amazing-job-1110d1779e0b

Six months ago I quit my job as a junior JavaScript developer and travelled around south east Asia for five months. Within a week of getting back to the UK, I had three job offers and had accepted an offer for almost double my previous salary. It wasn't easy, but it was worth it. Here's how I did it.

The Aim

I was planning to travel around south east Asia for 3–6 months, and I knew that I wanted to get a better job when I returned. My junior JavaScript job had shown me three things:

- I loved to code, not only as a hobby but as a job.
- I enjoyed back-end development work as it was removed from the CSS and visual design of front-end development work.
- I didn't like Angular that much.

With this knowledge, I set about creating a target for where I wanted to be by the time I finished traveling, and what I needed to learn to get to that point.

My target was to be a **mid level full stack developer**.

The Recon

To become the mid level full stack developer I wanted to be, and to find out where I needed to improve, I looked for jobs that I would actually want. I saw what they were requiring in a candidate and what they said was desirable.

Through this research, I found a list of things that most of the jobs required:

- Strong understanding of JavaScript
- At least one framework (usually React or Angular)
- Ability to create a REST API (usually express)
- Knowledge of one or more database (usually MongoDB or SQL)

Formulating the Plan

With this information, I knew that I needed to up my game in a few key areas:

1. I needed to finish reading *You Don’t Know JS*. This book series helped me really understand the basics of JavaScript, so I reasoned that it would be great for learning the more advanced stuff.
2. I needed to become very good at one front end framework. I looked into Angular, Vue, and React and decided on React. I’d done a bit of React before and liked it, I wasn’t a fan of Angular, and Vue wasn’t requested in many job adverts.
3. Although I could just about create an express API from scratch, I wasn’t very confident and could only do the basics. I needed to up my game across the board with this, especially if I was wanting to have a job that prioritized back end work.
4. I could either stick with relational databases (postgreSQL) or learn a new DB. I decided that I would go for learning MongoDB as it was massively popular and added NoSQL databases to my CV.

I wasn’t sure how long I’d be traveling, so decided to plan my learning out like this:

1. Learn React (and Redux) while reading *You Don’t Know JavaScript*.
2. Get much better at express and learn MongoDB.
3. Make something with the skills I’d learned.
4. Learn other things.

I did this for a very good reason. If I ended my travels after three months and had completed #1 but was still part way through #2, then I would still be able to apply for mid-level developer jobs — they’d just be limited to mainly front-end jobs.

If I did back-end learning first, after 3 months I still wouldn’t be very confident with any front end framework. That would probably be a sticking point for every mid level job. I could try to apply for a back-end only developer role, but they all seemed to want much more experience than I had.

Learn React and Redux, and read *You Don’t Know JS*

I'd done a little bit of React before, using Youtube videos and normal web tutorials, but this time I wanted to make sure that I was learning it properly. This led me to getting the *React JS and Redux: Mastering Web App* course from Udemy.

This course was brilliant, and the little bit of experience I had with React helped a lot, but I had no experience with Redux. By the end of this course, I felt like I could have built a pretty complex website and was confident with both the syntax and formats of React and the data control of Redux.

This course took me about 2 weeks to complete, but I was working on it for an hour or 2 once or twice a week. I was on holiday in Asia so this wasn't my main priority.

As well as working on the course when I had my laptop, I read my way through the last few books in the *You Don't Know JS* series. I stored the online Github version on my phone and read it while I was resting between climbs.

Having gotten some real world experience, I understood a lot more than the last time I tried to read these books. Things like Promises are really hard to understand if you’ve never used them.

Get much better at express and learn MongoDB

Now I felt comfortable creating front ends it was time to work on my back end skills.

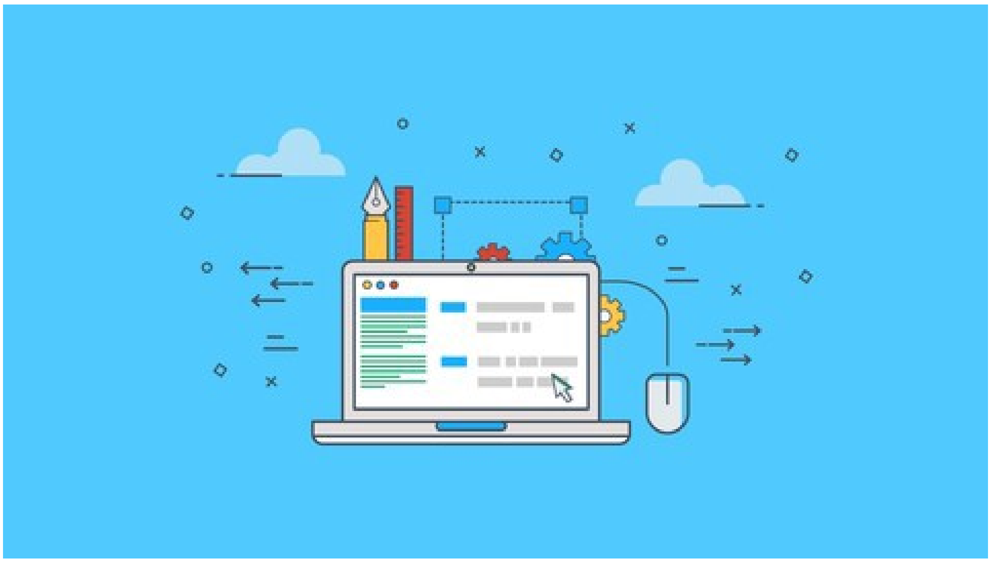
Again I decided to do a Udemy course. I found that because they’re complete packages, it progresses nicely and tends to cover the whole process. Of course you can learn all of the same stuff for free on Youtube or other online tutorials, but they never seem to cover a whole topic quite as well.

The next question was which course to do. There are free ones and paid ones, short ones and long ones.

I decided to invest in myself and buy a course ($10 is a tiny investment for such a lot of information) and I managed to narrow it down to about 3 courses. They varied in length between 7 hours and 25.5 hours.

I was tempted by the 7 hour course — I could get it done in a week or two whilst still enjoying my holiday. The 25 hour course would obviously take much longer, but seemed to cover more topics. By this point in the trip, I’d decided that I was going to be traveling for as long as by bank balance allowed, so I went with the 25 hour course.

Node with React: Fullstack Web Development



I’ve actually done a full review of this course here, and it’s a great course.

Having done the React and Redux course before, the React stuff was being repeated. So you don’t waste time, you can just skip videos you already know or put the video on 2x speed and “skim watch” for a refresher or to check for any bits of useful advice.

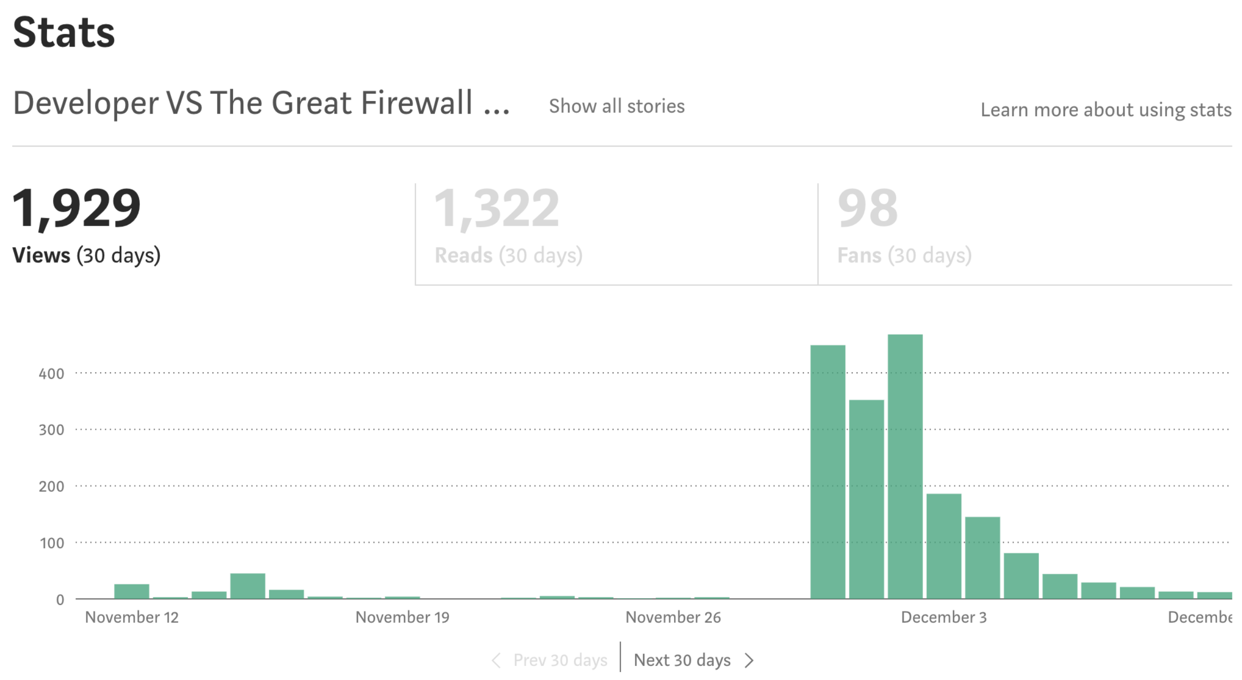
The back-end stuff is where this course excels. It covers a huge range of topics including: oAuth, email handling, payments, API key handling, MongoDB, mongoose, and advanced deployment.

The reason that this course is so long is that it covers **so many topics** in **such great detail.** This is good and bad, but I ended up watching most of the videos at 1.5x speed and pausing it when I needed to.

This course took longer than the first, but I managed to finish it in about 1.5 months.

I started Writing

At some point, while staying in China, I decided to write an article about how awful trying to do development was in China. It got accepted into the freeCodeCamp publication, and now has 1.6K reads and 791 claps. I was amazed that people read what I wrote and liked it.


Stats

Developer VS The Great Firewall ...

1,929 Views (30 days)

1,322 Reads (30 days)

98 Fans (30 days)

The Stats for the first 30 days of my article. Can you work out when it got put on Free Code Camp?

I decided that I was going to try to write more articles about what I was doing and cool things I find. This also ended up giving me a great talking point when it came to interviews.

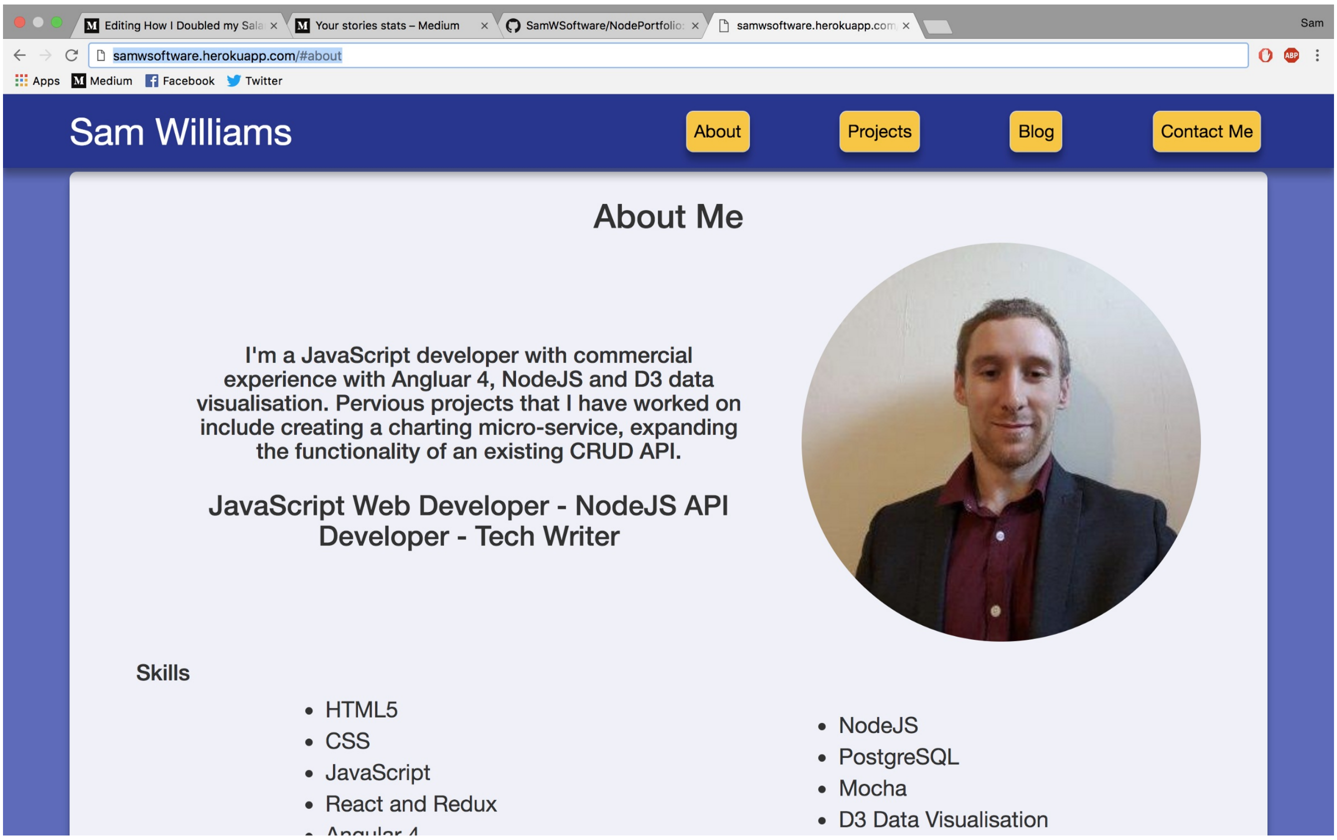
Writing a blog is also a great way to log your progress and allows you to see how you've improved.

Make something with the skills I'd learnt

Now that I had all the tools I needed, I wanted to create something with them. I was three and a half months into my travels and expected that I would be able to travel for another one-three months. This meant that I might need to start thinking about jobs soon if I wanted to start ASAP when I got back.

To apply for jobs, I needed to update my CV and portfolio. This gave me a great chance to use what I'd learned to build a new portfolio using Node and React.

This is exactly what I did. This is the result. As I said before, I'm not a designer, but it works well. I even used a Wordpress API to host blog posts on the page.


Editing How I Doubled my Sala
Your stories stats – Medium
SamWSoftware/NodePortfolio
samwsoftware.herokuapp.com
Sam
samwsoftware.herokuapp.com/#about
Apps
Medium
Facebook
Twitter
Sam Williams
About
Projects
Blog
Contact Me

About Me

I'm a JavaScript developer with commercial experience with Angluar 4, NodeJS and D3 data visualisation. Pervious projects that I have worked on include creating a charting micro-service, expanding the functionality of an existing CRUD API.

JavaScript Web Developer - NodeJS API Developer - Tech Writer

Skills

- HTML5
- CSS
- JavaScript
- React and Redux
- Angular 4

- NodeJS
- PostgreSQL
- Mocha
- D3 Data Visualisation

I knew that the design was awful, so I posted it in a web design review subreddit. I got lots of good advice, most of which revolved around spending the next few months researching and learning the basics of web design.

One bit of advice I did take was to utilize the bootstraps that existed. This resulted in a completely new website, using none of the skills I'd used before, but it does look less garish.

Applying for Jobs

The next place I visited was a remote climbing area in Laos. I assumed that this meant that there was minimal access to power and no wifi. This lead me to leave my laptop in Bangkok for the two weeks I was in Laos.

When I got to the campsite in Laos, I found out that they had 4G! I bought a local sim card (£2 for 1.5GB and then 89p for each extra GB) and had decent internet access. By this point I knew that I had two weeks in Laos, then about four weeks in China again before heading home. Time to get applying for a job.

Progress

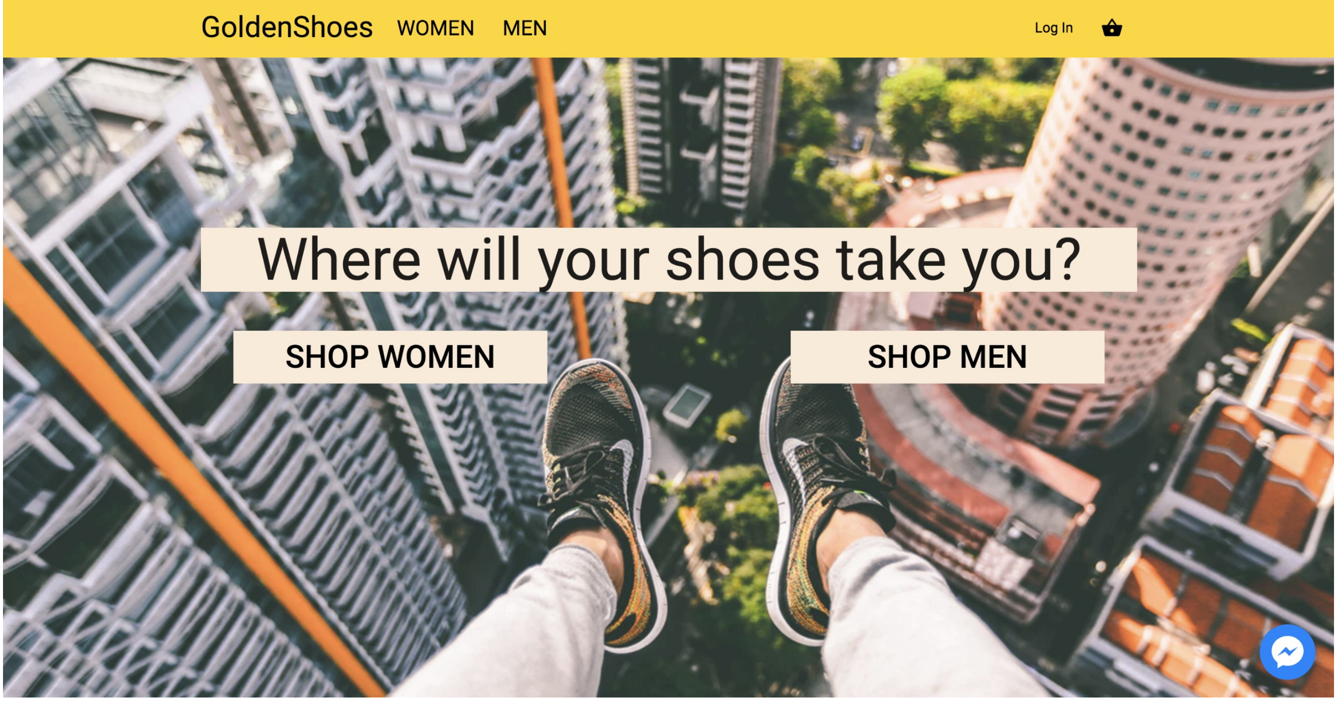
Over the course of the next one and a half months I applied to well over 50 jobs, had email and LinkedIn conversations with about 25 people, and managed to secure five job interviews for the week after I got back. Most of my interactions were by email, but if someone wanted to call directly then I tried my best to accommodate that (I was eight hours ahead).

I think that having an actual conversation is a great way to improve the relationship between you and the other person, increasing the chances that they'll invite you to interview or recommend you for a job.

If you don't feel comfortable having a call or Skype with people then you can stick to emails, but I think you're missing out on improving your chances.

Build More Things

One of the interviews asked me to prototype what I would do for a company with an awful website and minimal customer service. This was cool as it pushed me into learning about prototyping. I wrote a series of articles on how I prototyped and fleshed out the website. You can see the website here read my articles here.



The last thing I did on this was to create a chat bot for returning orders. I had to learn about the workings of chat bots and it took me about a week to get a decent working version. This was a massive factor in me getting the job.

Interviews

Now that I had these five interviews lined up, I had to convert them to job offers.

Preparation

I’ve written articles about preparing for a JavaScript interview and mastering the interview process. I made sure I ticked every box.

I knew all the points I was wanting to emphasise, how to answer the questions about my weaknesses and what each company did.

The Interviews

I’ve always been good at interviews, even though I’m naturally introverted. I’ve had to teach myself to act more extroverted and engage with the people interviewing me. If you find yourself acting shy and reserved in interviews, I recommend finding a book on emotional intelligence and ways to act more confident.

I had tech tests at three quarters of the interviews (having done the other two online) and my preparation paid off. There were small slip ups, but I talked through my thinking and finished all of the tasks well.

The Offers and Negotiating

From my first three interviews, I had two offers. After the fourth interview I knew that they weren't the company for me, so let them know I wouldn't be considering an offer from them. If you are ever in a similar situation, try not to lead them on if you know you're not going to accept an offer from them.

Now I had offers from two companies who were very keen to hire me. It was time to negotiate. I've never been great at negotiating or asking for more, but I read a few articles and went for it.

> You might think to yourself: "*well, I don't want to set high expectations, and the offer is already generous, so I ought to just take it.*"
> **No. Negotiate.**

> Or maybe: "*I don't want to start off on the wrong foot and look greedy with my future employer.*"
> **No. Negotiate.**

> *"But this company is small and— "*
> **No. Shut up. Negotiate.**

From Ten Rules of Negotiating a Job Offer. I recommend reading it if you're applying for jobs.

I definitely didn't do this perfectly (or even that well), but I did do a few things that strengthened my hand. I had two equal offers, but I had a preference for one company. I wrote myself a little script and (after some breathing exercises to calm myself) I called the director of my favored company.

> Hi Dave. I'm calling to give you an update on where I'm at. Thank you for giving me an offer, I'm glad you thought that I would be a a great fit in the company too.
> I've had another really strong offer and I'll be making my decision this weekend, letting you know by Monday at 12pm.
> There is one thing that could really strengthen the offer. I'm going to be relocating to wherever I take a job, and some assistance with the relocation costs would be amazing.

It didn't quite end up going exactly like that, but it helped a lot. I thanked them for the offer, told them a deadline for making the decision, and asked for an improvement in the offer without just asking for a higher wage.

Summary

I found out what I needed to get the jobs I wanted. I used this to create a plan and then used paid online courses to speed up the learning process.

I applied to loads of jobs and got five interviews. I negotiated the offers from the two companies I liked the best.

I've been working at MissionLabs for 2 weeks now and I'm loving it!