



Home [Subscribe To Podcast](#)

[Episodes By Date](#)

[Episode By Theme](#)

[About](#) [Donate](#)



CORECURSIVE #087

JSON vs XML

With Douglas Crockford

▶ LISTEN NOW

▶ PODCAST PLAYER



◀ [Sun's Mobile Blunders](#)



EPISODES

JSON vs XML



0:00:00

49:59:59

- ▶ JSON vs XML Apr 03, 2023 49 min
- ▶ Sun's Mobile Blunders Mar 02, 2023 51 min
- ▶ Shipping Graphing Calculator Feb 02, 2023 46 min
- ▶ The Unfulfilled Engineer Jan 02, 2023 42 min

This is foundational stuff for the web, and Doug is an iconoclast.

Guest


 Douglas Crockford **Douglas Crockford**
Pioneering Web Developer

Table Of Contents

- 00:00 - Intro**
- 02:16 - Introducing JavaScript**
- 03:06 - Fighting Microsoft**
- 03:54 - HyperCard**
- 05:06 - A Compromise**
- 06:33 - Electric Communities**
- 08:31 - The JVM**
- 09:20 - The Palace**
- 10:40 - Turner Broadcasting**
- 12:47 - Building With JavaScript**
- 14:34 - Dot Com Bust And JSLint**
- 16:10 - comp.lang.javascript**
- 18:30 - Starting Again**
- 20:36 - Raising Money**
- 22:05 - Selling Single Page Web Apps**
- 23:09 - Making JSON a Standard**
- 26:23 - Message in a Bottle**
- 27:41 - The AJAX Revolution**

32:26 - MIT

▶ JSON vs XML

📅 Apr 03, 2023

🕒 49 min

33:25 - Web 2.0 and The Conference Circuit

▶ Sun's Mobile Blunders

📅 Mar 02, 2023

🕒 51 min

37:22 - The FBI

▶ Shipping Graphing Calculator

📅 Feb 02, 2023

🕒 46 min

▶ The Unfulfilled Engineer

📅 Jan 02, 2023

🕒 42 min

43:58 - Embrace New Paradigms

45:30 - The Next Paradigm

47:56 - Outro

48:49 - JavaScript Frameworks

Transcript

Note: This podcast is designed to be heard. If you are able, we strongly encourage you to listen to the audio, which includes emphasis that's not on the page

Intro

Adam: Hi, this is CoRecursive and I'm Adam Gordon Bell. Each episode is the story of a piece of software being built.

I remember a final project in university and undergrad that I worked on. It was a semester long, self-directed class, and it was one-on-one with my advisor, my professor. I remember being in his office and going over the code I was working on and showing him all I'd written and being so proud. This was in 2004, and my project was Enterprise Java Beans. It was written in Java, but really it was written in XML. There was XML config for ANT and there was XML for Tomcat deployment and for ORM configuration and XML described how the services communicated and there were schemas that validated the XML. There was XML logging for config. And honestly, I think he was confused about what it was all for.

And I tried to explain to him what I had learned online, that it was easier to change and you didn't have to recompile things, and it was human readable, but machine verifiable and my project worked in everything. Originally, he had wanted me to do something related to search heuristics, but I



EPISODES

JSON vs XML



0:00:00

49:59:59

| | | |
|--------------------------------|----------------|----------|
| ▶ JSON vs XML | 📅 Apr 03, 2023 | 🕒 49 min |
| ▶ Sun's Mobile Blunders | 📅 Mar 02, 2023 | 🕒 51 min |
| ▶ Shipping Graphing Calculator | 📅 Feb 02, 2023 | 🕒 46 min |
| ▶ The Unfulfilled Engineer | 📅 Jan 02, 2023 | 🕒 42 min |

But today's show is about somebody who saw XML and said what my professor was too polite to say. Today is Douglas Crockford, and he's going to share the story of JSON and his discovery of JavaScript and its good parts, and his approach for finding a simple way to build software and his battles against XML and against complexity, his battles to say that there's a better way to build software. This is foundational stuff for the web, and Doug is an iconoclast for sure. Let's start with JavaScript.

Introducing JavaScript

Douglas: The first time I saw JavaScript when it was first announced in 1995, I thought it was the stupidest thing I'd ever seen. And partly why I thought that was because they were lying about what it was.

Originally it wasn't supposed to be called JavaScript. It was going to be called Moca, and there was a tension between Sun and Netscape. So Netscape was the browser maker, I don't know if people still remember them, but they were the first big commercial browser. So Sun had been claiming that if you write to the Java virtual machine, it doesn't matter what operating system you're running on, and that means we can be liberated from Microsoft. And Netscape said, "If you target all of your applications to the browser, the browser can run on all of the operating systems, so you're no longer dependent on Microsoft."

Fighting Microsoft

Adam: Microsoft was the big bad guy, both hated and feared.

Douglas: So they decided to have an alliance, and the first thing they agreed on was that Netscape would put Java into the Netscape browser, so they did, in the form of applets. So you could write applets in Java and they would run in the Netscape browser. Do you remember applet?

Douglas: Yeah, they were absolutely horrible. The next thing Sun demanded was, you have to kill Mocha which I think by that time had been renamed LiveScript because you're making this look like this stupid looking thing called LiveScript. Why are you doing that? This is just confusion.

HyperCard

Douglas: They remembered something that Apple had made, called HyperCard. HyperCard was this very nice interactive scriptable system that Bill Atkinson had made and was distributed for free for a while, on every Macintosh. It was all event driven, all scriptable, with a built in graphic editor and program editor. It was this really nice thing, and a lot of people started programming in this thing and were writing in a style of programming that the professional programmers of the day thought was impossibly hard, which was doing stuff based on events.

At that time, all the professional programmers were blocking on input and they couldn't understand how you could do anything else, but the beginners didn't know that it was hard, and so they were doing amazing stuff. Some of that stuff eventually turned into best-selling products, like the Myst series, came out of HyperCard. So Netscape thought they could do a similar thing for their navigator browser that, if they could get people programming in the same way that they did on HyperCard, on the browser, but now they can have photographs and color and maybe sound effects, it could be a lot more interesting, and you can't do that in Java.

A Compromise

Adam: But Sun was not happy about this. They said, "We thought we agreed that Java was going to be how you script the web." And Netscape probably said, "Listen, we can't rebuild everything to make it centered around the JVM. That's too much work and this scripting thing, we have works and is beginner-friendly."

Douglas: And so, they were at an impasse and their alliance almost broke when someone, it might have been Marc Andreessen, it might have been a joke, suggested that they changed the name of LiveScript to JavaScript, And we'll tell people it's not a different language, it's a subset of Java. It's just this little reduced version of Java, it's Java's stupid little brother. It's the same thing. It's not a

they lied to the world about what JavaScript was, and that's why the language has this stupid

▶ JSON vs XML

📅 Apr 03, 2023

🕒 49 min

confusing name

▶ Sun's Mobile Blunders

📅 Mar 02, 2023

🕒 51 min

Adam: Doug knows all this in retrospect, but at the time, all he knew was, here's this new scripting

▶ Shipping Graphing Calculator

📅 Feb 02, 2023

🕒 46 min

version of Java

▶ The Unfulfilled Engineer

📅 Jan 02, 2023

🕒 42 min

Douglas: So I took their explanation of what the language was and I read the spec or what material was available at the time about it, and I thought, "This is appallingly stupid. I don't know how they could do this."

Adam: This was in the mid '90s by the way, and Doug was working at a startup called Electric Communities. Many devs felt the same way about JavaScript, "This isn't Java, what subset is this? Who would use this?" Meanwhile, Doug's team was building something really cool.

Electric Communities

Douglas: We had built this amazing 3D world that you could interact in. You could have an avatar and you could customize the avatar and you could go walking around and teleport to places and encounter people there and talk to them like we're talking and do things. You could exchange objects and play games or build things or buy things or whatever it is you want to do.

Adam: This was like Second Life or the Metaverse, but years before that, they had been working on it all throughout the '90s, before the web was really much of anything.

Douglas: We were way, way, way too early to do this stuff. And the reason you've never heard of it is because we were way, way, way too early.

And we looked at the web, I looked at the web and I said, "The web is deficient in all of these ways, in obvious ways." It wasn't interactive. It didn't give you enough control over the visual presentation. It was not secure, that's a really big one there a lot of things that it couldn't do that still doesn't do very well. One of my big predictions that was completely wrong was that the web was not going to make it, because it was just so poorly designed.

Adam: For the online thing they were building, the monetization idea was to sell it to other companies.

Adam: Electric Communities was a Bay Area of BC funded social network.

▶ JSON vs XML

Apr 03, 2023

49 min

▶ Sun's Mobile Blunders

Mar 02, 2023

51 min

and something, we were catty-corner to Apple. Next door was Javasoft. There were a lot of tech companies in that area. Well, Stevens Creek Boulevard that's where we were. We were about 100

▶ Shipping Graphing Calculator

Feb 02, 2023

46 min

▶ The Unfulfilled Engineer

Jan 02, 2023

42 min

we were a fairly big company.

The JVM

Adam: Their online metaverse thing ran on the JVM.

Douglas: And that was my decision, and I hope it's the worst decision I will ever make in my life. I'm pretty confident that it is, but I still got some years left.

Java was problematic for lots of reasons. One was, it wasn't secure enough. We had a set of requirements for security and Java did not need it, but we thought that was okay, because Java was open and so we could fix it, but then they reneged on the open thing and then we were trapped. So that was difficult, trying to make it secure within the restrictions that they put on us.

Adam: One of the big problems was startup time.

Douglas: So it took us three minutes to load the environment. Java has a class loader and it incrementally brings each class in and verifies it and puts it in memory, And we kept hoping that we'd get better and it didn't get better.

The Palace

So we decided to park it and we pivoted. We acquired another social media company called The Palace and started trying to figure out how to put ads on the palace in order to make money.

Adam: The Palace was a program you downloaded, and then you connected to a palace, which was like a server, but presented as a series of graphical rooms all around a certain topic. It was like a graphical chat room. This is around the year 2000 and everybody wanted to be online, but the web was static, and this was a attempt to make it more interactive and social. Fans of Korn or South

Douglas: The difficulty we had was that advertisers were really skittish about putting their ads next to user generated content. What if we put up an ad and people saw bad things about us? What are we're going to spend it somewhere else.”

- ▶ JSON vs XML Apr 03, 2023 49 min
- ▶ Sun's Mobile Blunders Mar 02, 2023 51 min
- ▶ Shipping Graphing Calculator Feb 02, 2023 46 min
- ▶ The Unfulfilled Engineer Jan 02, 2023 42 min

not soon enough for electric communities. And so, since the Palace wasn't bringing in any money, they decided to try to do a consultive approach. They could build online experiences for other companies. So they took on a project from Turner Broadcasting.

Turner Broadcasting

Douglas: We'd had a long relationship with them, and they were going to do a children's collectible tradable card game that would run on the web, and they wanted us to work on it. They knew about the Palace technology that we had, and we said, "Yeah, we could do that." And so, the first question was, how do we put the Palace technology into the browser? One way would be to make it a Java applet, and in fact, we'd already done that twice, and they both turned out really poorly, was not happy with the results. So I didn't want to do it a third time.

I didn't think we could meet the quality requirements that Turner had if we did that. Somewhere, and I still don't know where I got this crazy idea that we could do it in the browser, just with JavaScript. So I got Dave Flanagan's book and read through it and then wrote a little demo page in which I had cartoon characters and you could drag and drop them around on the screen. Nobody had ever seen that before. I hadn't done anything that wasn't possible, because it clearly was possible, but it wasn't widely known that JavaScript could do that. So we showed that to the people at Turner and they said, "Oh yeah, that's exactly what we want." And they also liked that you didn't have to hit install anything. With the applets, you had to agree to an installation, but they had kids there and there with the COPPA laws now, there were restrictions on what you can ask kids to do on a computer network.

Adam: Turner Broadcasting is convinced, now Doug just has to tell his team.

Douglas: I had a team of some of the best Java developers in the world, and I said, "Okay, I'm back from Atlanta, and they've agreed this is the way we want to do it. So we're going to write this thing in JavaScript." So I asked for a show of hands, who wants to write in JavaScript? And then they said, "What's plan B?" Because none of them believed that this was going to work. The

JavaScript. Even the people who were writing JavaScript, hated JavaScript. It had no friends. So in order to get it done I had to do the coding myself

▶ JSON vs XML

📅 Apr 03, 2023

🕒 49 min

▶ Sun's Mobile Blunders

📅 Mar 02, 2023

🕒 51 min

▶ Building With JavaScript

▶ Shopping Graphing Calculator

📅 Feb 02, 2023

🕒 46 min

▶ The Unfulfilled Engineer

📅 Jan 02, 2023

🕒 42 min

Adam: So Doug sits down with Dave Flanagan's 1996, JavaScript: the Definitive Guide, and he gets to work building the front end of Turner's online card game. His Java developers never came around, but eventually he brought in some consultants who knew JavaScript and they got it done, and it was on time and it was on budget, and there was no Java applets installed at all.

Douglas: And in the process, I learned a lot about how JavaScript works, and eventually I had an epiphany that JavaScript was not a subset of Java. It is an entirely different language and has good parts that go way beyond what Java's able to do. The most important one is that it has functions, it has first class values with Lexical closure, which was a brilliant idea that we got with Scheme, which still had not made it into the mainstream. Turned out JavaScript was the first language to give us lambdas, and that was an amazing breakthrough. And once I understood that, my relationship with the language changed completely. I wrote an article called JavaScript, the World's Most Misunderstood programming Language, and began a career of trying to convince people to reconsider the language and use it in a more effective way. It turns out it, well, it's a multi paradigm language, but the important paradigm that it had was functional. We still haven't, as an industry, caught up to functional programming yet. We're slowly approaching it, but there is a lot of value there that we haven't picked up yet.

Adam: This style of programming was not popular. OO was supposed to be the great hope for the future, but Doug started to think that maybe this was a game changer. Maybe Electric Communities were onto something, they could raise more money and start building a new interactive web, but as they started thinking about more fundraising, the .com bust hits and they have to start scrambling.

Dot Com Bust And JSLint

Douglas: It was a little bit brutal. I remember vultures coming into our offices and looking around, looking at the furniture and at the equipment and trying to scrape stuff off our carcass as cheap as they could. And we had an investor who offered us a bridge loan and was going to lead the next round of investment, but they reneged on that. So they ended up buying the company for the cost of the bridge loan. That was pretty painful. And that's hard, because you invite all these people to

Adam: So Electric Communities ends up going under, just as Doug has a new toy he is excited about JavaScript but the winding down process is slow

▶ JSON vs XML

📅 Apr 03, 2023

🕒 49 min

▶ Sun's Mobile Blunders

📅 Mar 02, 2023

🕒 51 min

Douglas: That gives you a lot of free time. So we've gone through most of the layoffs and just had to be around to sign things and let people in the building stuff like that

▶ Shipping Graphing Calculator

📅 Feb 02, 2023

🕒 46 min

▶ The Unfulfilled Engineer

📅 Jan 02, 2023

🕒 42 min

So I was familiar with a parsing technique that Vaughan Pratt had developed, called Top Down Operator Precedence and decided I'd like to see if I can implement this in JavaScript. And it turned out it was really easy, because JavaScript was very closely aligned to that kind of presentation. So I had this JavaScript parser, what can I do with that? I thought, "Well, I'll have it parse JavaScript and tell me what's wrong with it." Because at the time, programming in JavaScript was really difficult. We didn't have exception handling yet. So if anything went wrong, it would just crash. And so, you had to write programs that never failed, and that turns out to be very difficult.

comp.lang.javascript

Adam: So Doug built JSLint. It was just a simple Java program to point out where his other programs had issues, and then with still nothing to do but watch Electric Communities unwind, Doug starts spending his time on the JavaScript news groups.

Douglas: And every once in a while, someone would post something, "Here's some JavaScript code that I wrote. I don't understand why it doesn't work, can someone tell me." So I take their code and I'd put it in JSLint, and sometimes I go, "There it is", and I'd post it back at them. And sometimes JSLint couldn't figure it out and it looked like it should be able to. So I thought, "Well, why is that?" And eventually I discovered that there's certain syntactic forms which make it easier for bugs to hide.

So if you don't use those forms, then bugs are easier to detect and your programs are better. And so eventually, I came up with the principle that if there are two ways to do something, choose the one that's less likely to cause errors. And that turned out to be a wildly radical idea, that there was a lot of resistance to that. Some people took it as an insult, like, "Well, obviously a stupid person could use that feature incorrectly, but to suggest that I can't, that's just mean." A result of that was that JSLint taught me how to think about quality in coding, which was something I'd never really considered seriously before. So JSLint is much smarter than I am about coding.

Adam: One reason for this was that JSLint was aware of ambiguities in the JavaScript grammar.

- Douglas:** And you can use semicolons to disambiguate the grammar and make it clear as to what the intention is or to cause the bug to crash. Once I understood that I started teaching always use
- ▶ JSON vs XML Apr 03, 2023 49 min
 - ▶ Sun's Mobile Blunders Mar 02, 2023 51 min
 - ▶ Shipping Graphing Calculator Feb 02, 2023 46 min
 - ▶ The Unfulfilled Engineer Jan 02, 2023 42 min
- over time. I've found others like, don't use ++, because there are problem cases with that. Use +=1 instead, that seems to be much more reliable

Adam: So Doug built up a list of these and adds them to his Linter, and eventually Electric Communities shuts down and Doug has to move on to whatever's next. He knows exactly what he wants to do.

Starting Again

Douglas: So being in a startup, I think is the most fun I've ever had. It's really great. You get to choose all the people you're going to work with. You get to choose people who are really smart. Everybody is really engaged in the collective problem that you're all trying to solve together. You don't have any stupid suits getting in your way. It's great when it's working. It's great.

Adam: So for Doug and Chip Morningstar, another Electric Communities alum, the next step is obvious.

Douglas: We had this idea of taking all the things we'd learned at Electric Communities and trying to simplify it and recast it for the web. And so, I got this idea that we could do what are now called single page applications in browsers using JavaScript.

Adam: They call their company State Software and they start building a proof of concept.

Douglas: So Chip and I needed a way of getting the data between the browser and the server and I had this idea that we can format the data as JavaScript, and then the browser will parse it for us and deliver us the values and everything's great. So that'd be a whole lot faster than writing my own parser, particularly if the parser had to be for XML, because that's heavy and stupid.

Adam: XML was the new hotness at this time.

Douglas: It was very popular. All the hipsters were excited about XML. There were lots of big companies that were promoting XML, including Microsoft and HP and IBM and Sun and everybody. And they all had these enormous tool stacks that were there to make XML less awful to work with.

| | | |
|--------------------------------|----------------|----------|
| ▶ JSON vs XML | 📅 Apr 03, 2023 | 🕒 49 min |
| ▶ Sun's Mobile Blunders | 📅 Mar 02, 2023 | 🕒 51 min |
| ▶ Shipping Graphing Calculator | 📅 Feb 02, 2023 | 🕒 46 min |
| ▶ The Unfulfilled Engineer | 📅 Jan 02, 2023 | 🕒 42 min |
| ▶ Raising Money | 📅 | 🕒 |

Douglas: For me, the most difficult thing was raising money. You're constantly going to Sandhill and calling on people who don't understand what you're doing, and are looking to take advantage of you if you can, and they're going to do that, but you have to go on your knees anyway.

I found that stuff to be really hard, although some of them I really liked. And sometimes I'd be sitting in those meetings and I'd be thinking, "I wish I was rich enough to sit on the other side of the table, because what they're doing right now looks like a lot more fun than what I'm doing right now." And it was even more difficult raising money then, because at this point, the .com bubble had popped and all VCs had been hurt really badly by that. So they were only funding sure things at that time, in late 2001, early 2002.

And I thought we were a fairly sure thing, because we had already implemented our technology. And by this point, Chip and I understood the problem really well. And we had a new server and JavaScript libraries done in just a few months. And we had demonstrations. We could show the actual stuff. So it wasn't like we were raising money so that we could do a thing. We had already done the thing, we needed the money so that we could roll it out. And that wasn't enough for them. They wanted to see that we were already successfully selling it. And I was like, "If we could do that, we wouldn't need you."

Selling Single Page Web Apps

Adam: So State Software was only able to get one small investor, but Chip and Doug persisted in building their vision of the interactive web, and then they started trying to sell what they had built.

Douglas: And we explained how our system worked and that we had this JavaScript format that we use for moving the data back and forth. And some of our customers were confused and said, "Well, where's the enormous tool stack that you need in order to manage all of that?" "There isn't one, because it's not necessary", and they just couldn't understand that. They assumed there wasn't

- Adam:** It's like you had an electric car and they were like, "Well, where do we put the gas in?"
- ▶ JSON vs XML Apr 03, 2023 49 min
 - ▶ Sun's Mobile Blunders Mar 02, 2023 51 min
 - ▶ Shipping Graphing Calculator Feb 02, 2023 46 min
 - ▶ The Unfulfilled Engineer Jan 02, 2023 42 min
- we just committed to XML, sorry, we can't do anything that isn't XML." And there are some people who were a little bit more open who said "Well we can only do things if it's a standard." And they a standard."

Making JSON a Standard

Douglas: So I decided I will make this thing a standard. So first thing we did was pick a name, and the first name we picked was JSML. It's going to be the JavaScript message language, but it turned out there was a Sun thing called JSML. So we couldn't do that. So we thought about a minute and came up with JavaScript object notation. I managed to find, there is a web hosting company that was going bankrupt and they were selling off all the domain names that they owned, and json.org was one of them. So I snatched it up and I wrote a one-page website that described JSON, because you don't need much more than one page because it's really not very complicated.

Adam: That was it. That was the creation of JSON, which everyone is using today. But back then, everyone rejected it.

In some ways it was a marketing problem. On one side, you had Doug, trying to convince customers that they can build interactive applications on the web using JavaScript and this simple thing called JSON. But on the other side, you had XML that had these big companies behind it, IBM, Microsoft, and big consultants. And later they even had some tech influencers like Dave Winer.

Douglas: He's someone who should have known better. He had a website called scripting.com. His style of scripting came from a clever program that he had written for the Macintosh called Frontier, in which he had a scripting language and an outliner and a word processor and a database, all in one program. And the idea was that you could do virtually anything in Frontier with a little bit of scripting. And he was also one of the big promoters of SOAP, the Simple Object Annoying Protocol.

I don't remember what the A was, but it might have been atrocious or abominable, I don't know.

But SOAP was a big deal at the time. They were right in wanting simplicity. They didn't accomplish it, but they put simplicity in the name as sort of a conspiratorial thing. And so, when I started showing

- time he complained that, "this isn't even XML. We should find who did this in string them up now", which was a really ugly thing to say
- ▶ JSON vs XML Apr 03, 2023 49 min
 - ▶ Sun's Mobile Blunders Mar 02, 2023 51 min
- Fortunately, nobody listens to Dave Winer, so I'm still here.
- ▶ Shipping Graphing Calculator Feb 02, 2023 46 min
 - ▶ The Unfulfilled Engineer Jan 02, 2023 42 min

was realizing there was a powerful language there, when you didn't compare it to Java when you looked at it for what it was, but there was some people using JavaScript.

Douglas: Some of the people who were doing JavaScript, were doing it badly and didn't want to be told that they were doing it badly. There is this notion among second rate programmers that the most important thing they do is express themselves. It's not making programs that work well and are free of error. That's way down on the list. What's much more important is expressing themselves, that, I'm an artist and I express my arts by leaving out semicolons. It's that kind of thing. So what I was saying did not resonate with that at all. So that audience was not happy to hear it, and JavaScript was just not being used.









Message in a Bottle

Adam: And then, with what little funding they had and their troubles convincing customers, State Software had to shut down. The biggest thing they left behind in retrospect, was Doug site, json.org.

Douglas: It was like I put a message format in a bottle and threw it in the internet. I'd start working in television. So at that time, we were making the transition to digital television. Up until that point, we'd had analog television. And so, I was going to all the conferences on technology, on law and regulation and everything, and writing reports about my predictions about how the digital transition was going to go and when it was going to happen and what the standards were going to be.

Adam: So it seems like Doug has lost, or at the very least, XML and full page replacement has won. Doug's often exile, writing reports for cable TV companies, and XML is just getting more and more popular. And then XML requests are added to the browser.

And there's a big push to move from HTML to XHTML, but then Gmail comes out and it was like Doug's previous work. It was like the Turner Broadcasting JavaScript, but even more so. It was a single page app and it was snappy and it felt like a desktop experience, and it ushered in the AJAX revolution.

- Douglas:** Suddenly there are all these people writing for the browser who'd never been in the browser world before and they were being told that you had to use XML to transmit the data. In fact
- ▶ JSON vs XML  Apr 03, 2023  49 min
 - ▶ Sun's Mobile Blunders  Mar 02, 2023  51 min
 - ▶ Shipping Graphing Calculator  Feb 02, 2023  46 min
 - ▶ The Unfulfilled Engineer  Jan 02, 2023  42 min

Adam: This was a very pragmatic choice, but the XML gurus weren't going to move on so easily.

Douglas: So there is an intellectual trap in XML, that it's really complicated. It doesn't look like it should be complicated. It's just angle brackets, but the semantics of XML can be really complicated, and they assumed it was complicated for a reason. This problem of data interchange is so complex that there's a conservation of complexity, that your format has to be complex, and the tools supporting that format have to be complex, so that the applications are manageable. And that's a false assumption, there is no such thing as conservation of complexity.

That generally, if you push on simplicity anywhere, you can get it everywhere eventually. So simplicity is the thing you should be driving, not complexity, but they didn't know that. And I think if you have bought into that, you have emotional connection now to this format, and you've been telling people that it's complicated for a reason, and it's good that it's complicated. And then this little piece of syntactic fluff shows up, which is solving the same problems without any complexity at all. You look stupid. And also, you have invested a lot of time and energy into learning this thing, which turns out is becoming irrelevant. And that's a tough thing. And if you're a consultant, it's even harder, because you've established a standing in the community that you have clients because this stuff and this stuff is no longer the thing. And so, a lot of them took this really, really hard.

Adam: While all this was going on, while JavaScript was becoming more important and pragmatists were starting to investigate, JSON, Doug had mainly moved on, but then some of his friends from Electric Communities started calling him up.

Yahoo

Douglas: So they went to Yahoo while I was doing my consulting, and they called me up and said, "Hey, you should be working here." Because at that time, it looked like Yahoo was going to become an interesting company.

Adam: So Doug started working at Yahoo.

- ▶ **Douglas:** The Yahoo offices were in Sunnyvale, California, on the edge of the Lockheed Martin campus. I think they had four buildings there and several other buildings scattered around in nearby areas. ▶ JSON vs XML ▶ Apr 03, 2023 ▶ 49 min
- ▶ Sun's Mobile Blunders ▶ Mar 02, 2023 ▶ 51 min
- ▶ **Douglas:** My best job in the company. Basically, I didn't have any real responsibilities. ▶ Shipping Graphing Calculator ▶ Feb 02, 2023 ▶ 46 min
- ▶ The Unfulfilled Engineer ▶ Jan 02, 2023 ▶ 42 min

Douglas: In fact, my initial role was to teach the company how to write in JavaScript.

So I thought of myself as a consultant. I was still a consultant, but this time my client was a company I was working for. So I'd just go around and advise people. I would teach. The company had offices all over the world, so I would go all over the world, teaching everybody how to write JavaScript. There are lots of programmers there who wanted to use JSON, but hadn't up until that point. But when they saw that I was working for the company, they took that as a sign that it's okay. So they started doing it. And at that time, Yahoo was still a very influential company. So if Yahoo was doing this, then we can do this. And the whole thing started to slide.

YUI

Adam: In certain circles, JSON became the way that you talked to the client side, not XML, but Doug had more to say than, just use JSON. He also wanted people to use JavaScript properly – use semicolons, use a functional style, don't use a vowel, use JSLint and so on. Doug was an evangelist in the truest sense, and one thing he was evangelizing for was this open source Yahoo project called YUI.

Douglas: But the sites at Yahoo were not using it. So I tried going around to them and saying, "You should be using this thing." And they said, "Eh, we're using this standard stuff." So it was the problem again, okay, I need to make this thing a standard in order to get them to use it. So I went out and started teaching the world how to think about JavaScript properly so that they would want to use YUI. The idea that if they could see that the world was using YUI, then the company would use YUI.

Adam: It's a complicated way to sell something.

Douglas: Yeah, well, sometimes that's what you have to do. We didn't have the authority in management at Yahoo to say, "We will do the right thing." Instead, we had to trick everybody into

▶ Sun's Mobile Blunders

Mar 02, 2023

51 min

Adam: Some teams embraced Doug's style right away, and some never came around to this style, to the libraries he was pushing, or Doug, but that was life for him for a while. Travel around the world

▶ Shipping Graphing Calculator

Feb 02, 2023

46 min

▶ The Unfulfilled Engineer

Jan 02, 2023

42 min

was a powerful language. One stop he made was to MIT, because the language he had found in JavaScript was really a lot like Scheme. And if MIT embraced it, then maybe that could change things.

Douglas: I took it from the approach of, here is this poor little orphan, this is your child, and I'm bringing her home to you. This is yours. You should love her. And I remember as this grizzly old man with a huge white beard, I was describing how functions work in JavaScript and how they did closure and all that. And he said, "You can do that in Scheme." I said, Exactly."

Adam: MIT eventually embraced JavaScript. Years later, they would reissue the structure and interpretation of computer programs in JavaScript, but that was in the future. At this point, things for the web were just starting to pick up.

Web 2.0 and The Conference Circuit

Douglas: Forrester Research had predicted that the web was going to die within a couple of years, and it looked like it was going to happen. And then JavaScript was rediscovered, and suddenly the web became the big thing again. So in retrospect, maybe that shouldn't have happened, but it did. So we still have the web, but it's a better web than what it was.

Adam: Talking about the same subject month after month, Doug found he had even more to say.

Douglas: I would talk about something and I would get a thought and I'd next time I gave that talk, I'd put it in, and eventually the talk would be so big that I'd take chunks out and make a new talk about the chunks I removed, and then I would keep growing again, and another talk would come from it. So it became the mother of a lot of stuff that I talked about. One of my favorite conferences was The AJAX Experience, in San Francisco. That was the first of the important web conferences, and it was several days, going very late into the night, and everybody was there, and it was so much fun and we were all influencing each other. We were all trying to figure out what this was about.

SOAP. JavaScript started being taken seriously as a programming language and traffic to json.org

▶ JSON vs XML

📅 Apr 03, 2023

🕒 49 min

just went in but then some trouble struck

▶ Sun's Mobile Blunders

📅 Mar 02, 2023

🕒 51 min

▶ Online Booty Call

▶ Shopping Calculator

📅 Feb 02, 2023

🕒 46 min

▶ The Unfulfilled Engineer

📅 Jan 02, 2023

🕒 42 min

Douglas: So one day I got an email from my hosting service saying I was using way too much bandwidth and that they're going to put me in a more expensive bracket. And I thought, "Whoa, what? What's going on? I don't think I've suddenly become so popular. So what happened?" So I looked at my logs and I see I'm getting a whole lot of traffic from OnlineBootyCall, and they're loading json.org/json.js.

Adam: OnlineBootyCall is exactly what you think it is, and it still exists today. I'm not sure how popular it is, but apparently they were using JSON. And back then, to take a JavaScript object and turn it into JSON to send it over the wire, it wasn't built in. You had to bring in Doug's json.js file and include it, which had a stringify function you could use.

Douglas: So apparently I'd left a copy of json.js On my server and they were loading it from there, which is a really bad practice, because if you load some third party script from another server, they get all of the privileges that you have. So from a security perspective, it's a terrible thing to do, and they're doing it and they're costing me money now.

So I decide I have to stop them. So what should I do?

I wanted to do it in a way which wouldn't hurt anybody, but would just make it clear that something needed to get fixed. So I decided to add a statement to the beginning of JSON.js, which was alert of the string important, remove this statement before deploying;. So I pushed that, and now suddenly everybody goes to OnlineBootyCall, they get this alert, and the people at OnlineBootyCall see that, and they go, "Oops." And they fix it.

Adam: Doug's friend saw the change in json.org/json.js and asked Doug about it, and Doug told him the whole story, and his friend wrote it up and put it on his blog.

Douglas: And the guys from OnlineBootyCall posted a reply or a comment on Steve's site and said, "Whoa, we screwed up. Sorry, Doug was right and we shouldn't hotlink other stuff. We won't do that again."

- Adam:** So happy ending, except now Doug has to start keeping a closer eye on his server logs, because this website needs to stay up. JSON has never been more important but also things like
- ▶ JSON vs XML Apr 03, 2023 49 min
 - ▶ Sun's Mobile Blunders Mar 02, 2023 51 min
 - ▶ Shipping Graphing Calculator Feb 02, 2023 46 min
 - ▶ The Unfulfilled Engineer Jan 02, 2023 42 min

Douglas: And one day I noticed that JSON. Something else maybe ActionScript or something. Russia. I thought, "That's weird, but I know what to do." So I went to the file and I put this alert on it.

Adam: What should be happening is, all these visitors who are loading that file should be getting the alert, the same as onlineBootyCall and contacting whoever is doing the Hotlinking to fix it, but instead, nothing happens.

Douglas: I thought, "Whoa, that's really weird." So I made the file inert, nothing happened. Said, "Okay." I deleted the file and now I have error responses and retries.

Adam: However this file's being loaded, it clearly doesn't differentiate error types. It's getting a 404 and just retrying again and again, and that's just making it worse, adding more server load.

Douglas: So then I thought: I know, I'll navigate the page, I'll change the page's location and send them someplace else.

So I got these Russian guys, how can I teach them a lesson? And I thought, "I know, I'll send them to fbi.gov and they'll look into it and that'll frighten them so much that they'll stop doing this and leave me alone."

So I did that. Next day, all of my websites are down, nothing's working.

So I call up my hosting company and they said, "Oh yeah, you've had a security breach. Apparently someone got in and is using your site to do a denial of service attack against the federal government. And we're going through the system, trying to figure out how they accomplished that." And I said, "Oh, I did that. I'm sorry I didn't intend to fight the federal government, sorry."

Adam: So Doug was nervous, but the FBI never showed up at his door. The sites got restored, and the hosting company must have convinced the FBI that there was nothing to worry about, but JavaScript and JSON just continued to grow in popularity. And Doug wrote his book, JavaScript, the Good Parts. Eventually it became time to circle back to that original problem, the one that had held State Software back, standardization. So Doug went to the internet engineering task force and tried

The IETF and ECMA

▶ Sun's Mobile Blunders

📅 Apr 03, 2023

🕒 49 min

▶ Shipping Graphing Calculator

📅 Mar 02, 2023

🕒 51 min

▶ The Unfulfilled Engineer

📅 Feb 02, 2023

🕒 46 min

📅 Jan 02, 2023

🕒 42 min

Douglas: So they assigned a handier to me who definitely helped, gave me advice on which things were important and helped me with some editing. But overall, it was a really painful process. And in

Application/JSON, which was weird because JSON is not an application, it's a text format. It's a way of representing data in text. And I think maybe that... Well, I don't know why they've forced that on me. I'd like to think it's because there were some XML fans who were resentful about what I'd been doing and decided that I didn't deserve to have text that was reserved for XML. So they got text XML, but I got Application/JSON, which didn't affect me personally at all, but for the people who use it, it's not a big deal. It's a little ugliness in the header.

Adam: How long did this process take?

Douglas: It seemed like forever, but I'm sure it wasn't very long.

Adam: And then, I guess the whole process is drafting the document and then once they all agree on it, then they send out the actual [inaudible 00:41:19].

Douglas: They send it out and then it's a permanent document or as permanent as bits can be. Then later, they came back to me and asked me to be the editor on a project to make an RFC, which would be the JSON Standard, which is a different document. And that was even more painful. So in the end, I left IETF and went back to Ecma, which publishes the EcmaScript standard. And I found that to be a much nicer group of people. And so, we published Ecma 404, which is the official JSON standard.

Adam: And from there, JSON became an ISO standard. And so XML fans can't claim that JSON shouldn't be used because it's not a standard. But really by the time of standardization, and by the time Doug's book started to get popularity, the tide had already turned.

XML Lost and Doug Won

Douglas: One indication of that is, you can go to Google Trends and you can do a query for JSON, XML. And you see two lines, as for the whole history, which Google keeps back to about 2004. And you see XML at the top, declining, going to zero ^and you see this little JSON line going up, and they

- Adam:** So JSON survived and became huge, but I guess you know that, and Doug's approached the JavaScript because of his book JavaScript: the Good Parts. And because of all his early
- ▶ JSON vs XML Apr 03, 2023 49 min
 - ▶ Sun's Mobile Blunders Mar 02, 2023 51 min
 - ▶ Shipping Graphing Calculator Feb 02, 2023 46 min
 - ▶ The Unfulfilled Engineer Jan 02, 2023 42 min
- caught up to Doug. And so today, reflecting back on it, day by day, year by year, Doug says some parts were hard, but overall, he's had an amazing career.

Douglas: I've been all around the world, talking to wonderful people everywhere. Most everybody's happy to see me. That's always nice. So yeah, it's been wonderful. I've had a really good time and a few of the things I've done have become influential, and that's good too. I'm famous for a programmer. Programmers don't generally become famous, but I'm fairly well known. Once I had experience, I was on vacation in Hawaii, met my brother there for lunch. He lives in Hawaii. And a couple of guys walk up and ask, "Are you Doug Crockford?" I go, "Yeah", because they'd seen my videos. And my brother was really impressed by that, it's never happened to him.

Embrace New Paradigms

Adam: One of the lessons Doug would like to pass on is, don't be too tied down to the way you develop software. Embrace and explore new paradigms.

Douglas: I used to be a big advocate of Object-Oriented Programming. I used to sing in that church and I don't anymore. I think Object-Oriented Programming was an important advance over procedural programming, which is what we had been doing, but we're stuck. We're supposed to change paradigms every generation, and we haven't. So we're way behind now. Object-Oriented Programming hit the mainstream in 1980. We're in the 2020s and we're still there. We should have moved on.

So I'm not advocating for JavaScript anymore. I'm advocating for the next language, the thing that comes after JavaScript. And the thing that language needs to be able to do is secure distributed programming, because the world is distributed. Programs don't run in a single computer anymore. They run in lots of computers that are all interacting, all communicating. So it's not about computation anymore, so much as it's about communication. And we need languages which are built to solve the problems in that paradigm. And we don't, languages like Java and C++ are still designed in the Von Neumann architecture, where everything happens in one process on one machine. And that's not the way the world is anymore.

The Next Paradigm

Adam: So what's next?

▶ JSON vs XML

📅 Apr 03, 2023

🕒 49 min

▶ Sun's Mobile Blunders

📅 Mar 02, 2023

🕒 51 min

▶ Shipping Graphing Calculator

📅 Feb 02, 2023

🕒 46 min

Adam: Actors are all about message passing and failing fast. And they've had some success with

▶ The Unfulfilled Engineer

📅 Jan 02, 2023

🕒 42 min

Douglas: So for a while, I had ambition that I would be the author of the next language, but I have not seen any case where anyone who was successful with one standard, had a second hit. I've seen lots of attempts, like the guy who designed Pascal had a huge win with that, but he had intended it to be a teaching language, but it became a big application language, and it was not well suited to that. So he came back with another language called Modula, and then a sequel, Modula-2, which were much better languages, didn't take off. So it seems to me that you only got one chance, and I got my chance and I won with JSON. And so, I am not going to be the guy who gets to decide what the next language is. The best I can hope to do is to influence the brilliant woman or man who designs the next language. And so, when I go out speaking now, that's really who I'm talking to. I'm trying to inspire someone to make that language and to get it right.

Adam: And so, if you're out there, next language creator, and want to know how to succeed, here's some advice from a person who really did change how web programming is done.

Douglas: The success of JSON was totally serendipity. Getting the domain name definitely helped. There are some things that I didn't do that definitely helped. I didn't secure any intellectual property protection on it at all. I didn't get a trademark for the name or for the logo. I didn't get a copyright for the specification. I didn't get a patent for the workings of the format. My idea was to make it all as completely free as possible. I don't even require any kind of notice. No one has to say, "Thank you, Doug, for doing that." It's just free for everybody. And I think that definitely helped.

But what steps do you do?

I don't know.

Outro

Adam: That was the show. Thank you to Doug Crockford for your time and for sharing your story with us.



EPISODES

JSON vs XML



0:00:00

49:59:59

- There's so much of Doug's story I couldn't cover here. His early days are fascinating. He worked at Lucas Arts, he worked at the Skywalker Ranch, he worked at Atari, and Chin Morningstar, his State Sun's Mobile Blunders
- ▶ JSON vs XML Apr 03, 2023 49 min
 - ▶ Shipping Graphing Calculator Feb 02, 2023 46 min
 - ▶ The Unfulfilled Engineer Jan 02, 2023 42 min

Anyway, I'm going to share some of the details of that either on Twitter or in the next supporter episode or in the newsletter.

And yeah, if you want to support the podcast, join as a supporter. Follow the link in the show notes.

And wait, there's one more thing I wanted to find out from Doug.

JavaScript Frameworks

Adam: What do you think is the XML of today?

Douglas: I don't know. It's probably the JavaScript frameworks.

They have gotten so big and so weird. People seem to love them. I don't understand why.

For a long time I was a big advocate of using some kind of JavaScript library, because the browsers were so unreliable, and the web interfaces were so incompetent, and make someone else do that work for you. But since then, the browsers have actually gotten pretty good. The web standards thing have finally worked, and the web API is stable pretty much. Some of it's still pretty stupid, but it works and it's reliable.

And so, when I'm writing interactive stuff in browsers now, I'm just using plain old JavaScript. I'm not using any kind of library, and it's working for me.

And I think it could work for everybody.

Support CoRecursive

- ▶ project, some bug, or some incident with me. Apr 03, 2023 49 min
 - ▶ JSON vs XML
 - ▶ Sun's Mobile Blunders Mar 02, 2023 51 min
 - ▶ Shipping Graphing Calculator Feb 02, 2023 46 min
 - ▶ The Unfulfilled Engineer Jan 02, 2023 42 min
- Right now this is all done by just me and I love doing it, but it's also exhausting.
- ... recommending the show to others and contributing to the podcast are the biggest things you can do to help out.

Whatever you can do to help, I truly appreciate it!

Thanks! Adam Gordon Bell

Support The Podcast

About CoRecursive



A podcast about building software.

Favorite Episodes

[Subscribe to Podcast](#)

[Subscribe via Email](#)

[Follow @CoRecursive on Twitter](#)

[About](#)


Links:

JavaScript - The World's Most Misunderstood Programming Language

JSON


0:00:00 EPISODES ▶ JSON vs XML 60: 49:00

[About Episode](#) [JSON vs XML](#)

 Apr 03, 2023

 49 min


▶ [Sun's Mobile Blunders](#)

 Mar 02, 2023

 51 min

Updated Date: 03 Apr, 2023


▶ [Shipping Graphing Calculator](#)

 Feb 02, 2023

 46 min

[Download Audio File](#)

▶ [The Unfulfilled Engineer](#)

 Jan 02, 2023

 42 min

[RSS](#)

◀ [Sun's Mobile Blunders](#)



 author

NAVIGATION

[Home](#)

[Subscribe To Podcast](#)

[Episodes By Date](#)

[Episode By Theme](#)

[About](#)

[Donate](#)

[Podcast Rankings](#)

SOCIAL



EPISODES

JSON vs XML



49:50

Linked in

▶ JSON vs XML

Apr 03, 2023

49 min

▶ Sun's Mobile Blunders

Mar 02, 2023

51 min

Slack

▶ Shipping Graphing Calculator

Feb 02, 2023

46 min

▶ The Unfulfilled Engineer

Jan 02, 2023

42 min

LISTEN ON

Apple Podcasts

Spotify

Google Play

Sticher

Overcast

CoRecursive

© Copyright 2021

15

EPISODES

JSON vs XML

60

0:00:00

49:59:59