Hypermedia In Action

14. Conclusion

Hypermedia Reconsidered

I hope that in this book we have managed to convince you that hypermedia, rather than being a "legacy" technology or a technology only appropriate for "documents" of links, text and pictures, is, in fact, a powerful technology for building *applications*. In the books have seen how to build sophisticated user interfaces for both the web, with htmx, and for a mobile application, using HyperView, using hypermedia as a core underlying application technology.

Focusing in on the web, in particular, many developers today view the links and forms of "plain" HTML as bygone tools from a less sophisticated age. And, in some ways, they are right: there were definite usability issues with the original web. However, as we discovered while building Contact.app, by simply addressing four core limitations of HTML:

- By making any element capable of issuing an HTTP request
- By making any event capable of triggering an HTTP event
- By making all the different types of HTTP actions available
- By aking it possible to target any element in the DOM for replacement

We were able to build user interfaces that many developers would assume require a significant amount of client-side JavaScript, but using only hypermedia concepts.

The Hypermedia Driven Application approach is not right for every application (after all, no approach is right for *every* application) but, for many applications, the increased flexibility and simplicity of hypermedia can be a huge benefit. And, even if your application wouldn't benefit from this approach, it is at least worthwhile to *understand* the approach, it's strengths and weaknesses, and how it differs from the approach you are taking. The original web grew faster than any distributed system in history, and it is worth understanding the underlying technologies that made that growth possible.

14.1. Pausing, And Reflecting

The JavaScript community and, by extension, the web development community is famously

chaotic, with new frameworks and technologies emerging monthly, and sometimes even *weekly*. It can be exhausting to keep up with the latest and greatest technologies that are coming out, and, at the same time, terrifying that we *won't* keep up with them and be left behind in our career.

This is not a fear without foundation: there are many senior software engineers that have seen their career peter out because they picked a technology to specialize in that, fairly or not, did not thrive. The web development world tends to be young, with many companies favoring young developers over older developers who "haven't kept up."

We shouldn't sugar-coat these realities of our industry. On the other hand, we also shouldn't ignore the downside that these realities create. It creates a high-pressure environment where everyone is watching for "the new new" thing, that is, for the latest and greatest technology that is going to change everything. It creates pressure to *claim* that your technology is going to change everything, if you are trying to get attention. It tends to favor *sophistication* over *simplicity*. People are scared to ask "Is this too complex?" because it sounds an awful lot like "I'm not smart enough to understand this."

The software industry has tended, especially in web development, to lean far more towards innovating, rather than understanding the technologies of the past and building on them or within them. We tend to look ahead for new, genius solutions, rather than looking backwards to older ideas. This is understandable: the technology world is necessarily a forward-looking industry.

On the other hand, there have been a lot of great ideas in the past, many of which have been discarded. We are old enough to have seen hypermedia come and go as the "new new" idea. It was a little shocking to us to see it discarded so cavalierly by the industry. Fortunately, the concepts are still sitting there, waiting to be discovered and reinvigorated by a new generation of web developers. The original, RESTful architecture of the web, when looked at with fresh eyes, can address many of the problems todays web developers are facing.

Perhaps, following Mark Twain's advice, it is time to pause and reflect. Perhaps, for a few quiet moments, we can put the endless swirl of the "new new" aside, look back on where the web came from, and learn.

Perhaps it's time to give hypermedia a chance.

I hope you will.