The Open Source Way

Episode 14: Fundamental Library – An Open Source Library for Everyone

Transcript

Karsten: Welcome back to the Open Source Way after a summer break. This is our podcast series, SAP's podcast series about the difference that Open Source can be. And in each episode, we'll talk with experts about Open Source and why they do it the Open Source way. I'm your host Karsten Hohage, and in this episode, I'm going to talk to Fabian Tempel and Mladen Droshev about the Fundamental Library. Fabian Tempel is a front end developer who focuses on HTML and CSS. He is a web accessibility expert in the cloud integration accelerator's team. He is known, as I hear, as the CSS guru, and he joined Fundamentals as a contributor. Mladen Droshev is the architect and the product owner of Fundamental Library. He also used to work for the UI5 team for a while when they got started. Then he changed to Hybris, where they had free beer at the office and he's still with SAP after the acquisition, although SAP cancelled the free beer. And also, what we might want to know is that people call him Deno, and I may call him Deno throughout the podcast. Let's start with you, Deno. What is the Fundamental Library? What does it do?

Deno: Karsten, and hi, everyone, thank you for having us in this podcast about Fundamental Library. Fundamental Library, in the first place, is an Open Source library. Actually, there are a few libraries under the umbrella of Fundamental Library, but we'll talk about that later. The main goal is to help application developers to achieve a more aligned user experience, no matter which web framework they use.

Karsten: Okay, so before we go into any more details of what it can do and how it does that, who would use the Fundamental Library?

Deno: Anybody can use it, firstly because it's on Open Source, right, and then, well, it's targeting, obviously, Front-End. So, any scenarios internally, externally, such as partners or customer projects, can actually use Fundamental Library freely. And this basically allows them to keep using the front end framework of their choice or the one that they support and still try to align with the SAP design. And I can give you an example…

Karsten: Maybe before you give me an example, let's ask Fabian, because he joined as a contributor. So, at some point he started using the Fundamental Library. What's your example? Why did you start using it?

Fabian: My team is basically doing a lot of PoCs and frontrunner projects for cross integration topics at SAP, and we were looking for different technologies to achieve quick goals, to be honest, in our projects. So, I looked around for what was available in the market to align with SAP's design guidelines and to not utilize UI5 in the first place, because we wanted to try out different frontend frameworks and see if they were suitable for our team and our projects. And that's how I found out about Fundamentals. And since I like very clean HTML and markup, we went forward with this one.

Karsten: Okay, so in short, you wanted to use something other than UI5, but still wanted to make it look like Fiori UI5 in the end, basically, right?

Fabian: Exactly.

Karsten: Exactly. So, then Deno, does that mean the Fundamental Library is more something internal for SAP LOB subsidiaries, whatever? Or is it actually also of public interest, let's say?

Deno: It should be for everybody and I want to come back to Fabian. So, Fabian is a great colleague of ours. He didn't only use Fundamental Library, but he engaged and helped us improve the library as well. So, we had like both cases: As a stakeholder and also contributor at the same time. And we're targeting, as I said, internal applications, but also external, especially if you want to develop something in the SAP ecosystem. And in this case, you want to make it, you know, align with the other existing SAP applications you have on your site. It will make it easier for the user, you know, to keep using even the new application next to older SAP ecosystems.

Karsten: Okay, and how did that get started for you? That you became involved in that?

Deno: For us, this started back in the days at Hybris. It was around the time that SAP acquired the company, but it was still, I think, a more kind of independent unit. And we, as part of Hyrbis, started the conversations about a more aligned user experience. How that happened? So, initially, I was starting as a startup around the commerce or commerce applications and solutions. Each team had the freedom as a startup to basically decide what framework, on the backend, on the front end database to the site reflecting the customers and the users, right. Because, you know, Hybris believes that, you know, the team itself, the application team, knows best what customers need. And then after a couple of cycles, it ended up having very fragmented frontend with different applications not aligned at all. And when you go to customers, you know, you present your portfolio. It didn't look, you know, like a serious company, so we started the conversation about how we could make this more aligned without forcing the application teams to change their framework. And the change had to be very minimal. And this is where Fundamental started. So first, we started with a very thin, small library, CSS based with the HTML specifications to allow them to be more aligned.

Karsten: And also, the original motivation was to bring on board, as far as user experience is concerned, the acquisitions and LOBs from SAP. Right. But now it's also open as an Open Source project to contribution and for use by it, by partner projects, customer projects and everything.

Deno: You're right. So, it started as a, let's say, small Hybris topic, but at the same time SAP started a broader conversation around the LOBs. How can we do something similar in the SAP world? And this is where we basically jumped and joined the different LOB and SAP designs with the broader context around Fiori and SAP design.

Karsten: How exactly would you describe the relation to Fiori and to UI5 when looking at Fundamental Library, if you had to draw a picture with words?

Deno: Oh, yeah, initially, when we started, this was kind of a parallel project, right? Completely focused on the hybrids and commerce applications. But later, we joined the discussions with the SAP design and the different LOBs. So, we shifted to following the SAP design guidelines more, and that is the visual design. And then obviously because of that, you know, SAP UI5 as the technology that supports Fiori, we also started conversations with the teams there, and there are more things to come soon that we are working on with the SAP UI5 teams, for example how we can share more development artifacts so more things to be reused in both worlds.

Karsten: Okay, Fabian, what would you say, does it fulfil its purpose?

Fabian: It definitely does, because since we began using a Fundamentals, Deno and I are actually in a working group to get above the UI5 teams and architects to align on some common CSS parts or common HTML parts to reuse across those two frameworks and also to be reused for any other framework out there. Or if you don't want to even use a framework and only rely on HTML because you might only need to do a click dummy or a presentation for whatever reason, you could use those kinds of things. And that's why Fundamentals really does fulfill its promise there, because it's easy to use for everyone and easy to get started.

Karsten: And when you just said any other framework that's things like Angular Vue and so on, right?

Fabian: Exactly, and all those frameworks are out there, a couple of others as well, to basically enable stuff which isn't available to the public, wasn't available a long time ago in the process called web components. That's the thing. All the frameworks try to emulate and do it little bit differently. But still, you could use the basics of HTML and CSS to integrate in all those frameworks.

Karsten: Okay, and that is exactly what Fundamental Library does or how does it make all these others look like UI5, Fiori?

Fabian: There is the basic Library which we called Fundamental Styles. This only provides CSS and HTML parts of components, and you need to do the whole functional layer on your own, which is then translated into the frameworks. And also, there are parts of the Fundamental Styles which utilize the specific framework, like from angular in Vue, and those things are ported into those frameworks to reuse there directly.

Karsten: Okay, Deno, you as product owner, would you describe it in the same way, or would you add anything?

Deno: Oh, it's exactly what Fabian said, but maybe I'll try to look at it from our perspective and explain how we see it. It's not going to change what Fabian said. So Fundamental Styles: We divided the library into smaller libraries. Fundamental Styles is a CSS layer that can be used with any Web technology, including Web components, because it's just CSS, which works with any Web technology. It worked ten years ago. So basically, you can take our library, bring it ten years back and use it and apply to any solution. And that's one of the use cases, that if there is an existing application, that SAP maybe acquired two years ago and it's an extremely successful application. You know, customers want to want to buy it; it will cost you minimal effort to make it look like Fiori without forcing and without big refactoring. So, this is one use case, only the CSS layer, because there is no dependence. You take the library; you apply the styling, and you get the Fiori look or the SAP design look. And then the next layer of Fundamental Library comes for Angular, for React and for Vue, which is basically on top of Fundamental styles, but benefiting the Angular framework, you know, Angular is a framework. It's one of the most successful front end frameworks, so the most used. It's Open Source, the community is huge. There are thousands of developers contributing. And then basically they're building on top of this. So, we're standing on the shoulders of giants, right. We're building the interaction. Like, for example, you take a button, the look and feel come from Fundamental styles. What we carry is what we implement. It's the interaction. When you click, you know, we take care of this click event and translate it to the application developers for the data manipulation. That's in a very nutshell, but benefit from all the performance improvements that the libraries bring, all the development experience, all the flexibility, the community, the low level of entrance. So basically, you can involve yourself with the library in no time.

Karsten: Okay, so for me, I always I try to put it simply, level one would be make it look like Fiori and level two would be make it behave like Fiori. All right. Okay, so let's maybe turn to the aspect that it's Open Source. Is that a lively community? Do you see external contributions happening a lot or is it more a different purpose Open Source project?

Deno: It is an active community and, one example, it's like Fabian, right? He not only uses but contributes back. So, we are very happy that there are a lot of teams within the company and individual contributors outside of the company who not only use the library but contribute back. And this is the key success for us. You know, we are not just driven top to bottom, like we dictate what's going to happen. No, we are working together with different line of businesses and contributors.

Fabian: Because when you're working with Fundamental Styles, you might notice a bug or something like that or that something works differently in a different browser set up because nothing is more difficult than front end to behave the same in every browser and every device on the planet. So that's why when using the components and some CSS of the Fundamental style library, you may notice some issues That's why I opened a GitHub issue, because I found some things that I disliked or wanted to have changed. In the past two weeks, I found a bug for an existing component, so the team around the Fundamentals needed to change that so that we and our projects could use the Fundamental Styles Library a little better. So, contributing and getting bugs out of the way is the way forward.

Karsten: Is that mostly what people do? Is it finding bugs and fixing existing things, or is it possible to actually contribute new stuff to the Fundamental core or however you call it?

Deno: People feel proud to contribute to the Open Source libraries because their work is being recognized and they can show it to everybody. So, it's really rewarding, you know, to open a bug and to fix it in an Open Source library. This is one thing. And this is one kind of contribution. The other one is new ideas. New components, new features that, with a small team, we cannot come up with everything ourselves. So at least people can open a ticket and propose something or ask something or even, you know, open a bigger request. And the third option, which we see in most cases out there, is great ideas and improvements that you can add to your library thanks to the community. And I can give you examples. So, these people, to come to your library, first had an interest, you know, before they looked at it. They liked it, but they see that maybe not everything is great. Maybe you can improve something with the automation and versioning, anything strategic. So basically, half of the success of Fundamental Libraries, is thanks to this contribution, for example, automation and semantic version. Colleagues from SAP concur. They have experience with that, and they help us basically spread this automation and versioning across all the libraries under the Fundamental. This helps us speed up new features out there, communicate them better and test them in a more automated way that we knew before. So, we worked a lot as well. This is the biggest benefit, actually.

Karsten: Speaking of communicating them better, what in general do you do so that it's easy for people to get started with Fundamental and easy to start contributing?

Deno: First, we keep. learning from them. We are not perfect. That is even, I guess, a better way to communicate. There's still something they're missing, but we're trying to improve. And one channel of communication is the proper semantic versioning of the changes through NPM. This is the community approach, and if you use it, basically, it makes it easier for the community to adapt to your changes.

Fabian: I basically follow the Twitter account of Fundamental Styles, which keeps track of the latest and greatest releases of all. If there is a new release of the Fundamental Angular project, then there will be a notification about that or about any other bug fixes on Twitter. Also, there's quite a lively channel to follow on LinkedIn for any news regarding Fundamental Styles. These are my most used channels for getting updates even while working on a project or just for having a look at the NPM stuff if there is a version number to be increased or changes or reporting something back Deno, that there's something wrong. So that's what we use.

Karsten: Okay, so the buildup of knowledge is mostly community driven; so, no centralized trainings or "getting started" conferences or whatever?

Fabian: No, because Fundamental Styles itself is based on very basic standards, like HTML and CSS, so hopefully every developer out there should know these very well. It became a little bit difficult lately in Web development, I would say, because no one's focusing on those technologies anymore. But since those are the very fundamental technologies to be used, there is no ramp up time, as I would say, for using Fundamental Styles out of the box.

Karsten: Okay, how come you said it became a little difficult? Nobody focuses on these anymore because they're all using abstracted tools doing their development or...?

Fabian: It's mainly that the developers nowadays focus a lot on those JavaScript frameworks and all those frameworks spit out HTML with nobody really looking. And maybe that's also the reason why your application is a little bit slow or sluggish for some users because nobody's focusing and looking at what the browser really displays to the user, because the browser as an application is only capable of displaying three things to the browser. That's basically HTML, CSS and nowadays a lot of JavaScript and there's nothing more on the technology side that the browser can interpret for the user to show the different things. And since the development process has got a little bit more focused around those JavaScript frameworks, HTML and CSS get left behind a little, because most of the frameworks out there are dealing okay-ishly with those parts. But it can be better, and Fundamental Styles does it, definitely.

Deno: And I want to add something here: It's connected, for example, it's connecting with accessibility as a product standard. We as developers, you know, at least me as a developer, I prefer to be lazy. And if I can avoid things, you know, and start using a framework that you give me out of the box, a lot of things, it's fine. But that accessibility cannot be just left to somebody else. Accessibility is responsibility for everybody, and I'm not saying this just to make it sound great, we all have difficulties, and we want everybody else to treat us, you know, equally. So basically, we must help everybody there to use our software. And achieving accessibility, as I said, is responsible for everybody starting from the design when it comes to colors. Contrast, then when you go to web development, basically all these Web elements like button links of tables, they require some accessibility settings to make them accessible for people with difficulties. And then when you go to the application developers, there is another accessibility requirement and the need to clear. So, this kind of question of "is the framework handling this for me" is a wrong thinking. So, you need to understand accessibility and your contribution there. And Fundamental, actually library styles, is trying to address this by using semantic HTML elements and then adding the missing part from the accessibility.

Karsten: Okay, and accessibility is also, on the one hand, an important concept in user interaction. On the other hand, it's an SAP product standard with very defined requirements, how any product that is labeled as SAP should or must behave in the ways of accessibility. Is that easier or harder to insure in an Open Source run project?

Fabian: I would say it's the same for every project because accessibility on the Web is not an easy task, to be honest, in the first place, because there are a lot of things to consider, and you can't make it right for everyone on the planet. That's the one thing you need to have in mind every time. But since Fundamental Styles is based on the pure web standards like HTML, there are some standards from the WCAG to be incorporated. And our SAP products are mostly based on those kinds of standards, including some other standards, like our German "Verordnung" as well. But most of the standards are based on the WCAG guidelines out there.

Karsten: Let me ask you something in that context, sorry if I interrupted you, continue later. I personally got the impression over time that sometimes you have the dilemma between "is it accessible" and "does it look nice".

Fabian: Actually, that isn't something that I would say.

Karsten: Okay.

Fabian: There's always a way to give it the proper semantic markup and treat it like the browser would like to have it, because every element in the HTML spec incorporates accessibility stuff already built right into the browser. So, if you're using the right text for a button or for a table or for a link or whatnot, there is some meta information given to the browser which handles the element properly for people with screen readers or whatever technology they are using. And if you're not following those standards and using whatever element you want to use, then those Fundamental informations get lost all the way. And so, if you're designing HTML in the first place to make it look nice, then there could be some issues, too, of having an inaccessible control or web application, that's for sure. But if you're basically applying the semantic stuff and all your markup in any case then there is no way that it should look a little bit odd but is accessible or it looks nice but is inaccessible.

Karsten: From what you just said, I already got the impression that my impression that I just described is probably like from the old days or something when browsers didn't have this built in yet, that they would automatically recognize if somebody were using the screen reader or other special requirements.

Fabian: Yeah, there are some additional standards which SAP is also part of, to write down those things like ARIA. If the normal HTML elements can't handle the requirements of an accessible control, then those ARIA things kick in and help you make things accessible. But the first rule of ARIA is not to use ARIA in the first place.

Karsten: Okay, that sounds like a completely different topic. We wandered far into accessibility here anyway, which is not the main topic and is a wide, wide field of course. Back to Fundamental Library: How and where would someone get started with Fundamental?

Deno: There are two ways to get started, one is to use it and the other one to contribute, and we are looking for both, right. If you want to use it, it's very easy. We publish NPM packages out there, so whoever in the JavaScript world would know how to consume NPM. So, you can google "NPM Fundamental Library" and you will find all the packages there, and they will lead you to a little more documentation. But just importing the entire package is good enough and you're good to go to contribute. In GitHub you can search for Fundamental Library and it will, you know, point you to the libraries and our GitHub repositories, the external ones. And then we have "Get started: How to contribute to the repositories".

Fabian: Fundamental provides an internal and, as far as I know, an external Slack channel as well. Developers who want to get a contact, there's a Slack channel and a Twitter account where you can ask stuff or call for help

Karsten: Let's come to my usual final question. If you could have three key takeaways that everyone who listened to this should remember. What would they be?

Deno: First one, Fundamental Library is trying to help frontend developers to keep using their framework of choice and still align their user experience online with the SAP design, for example. Second, yes, we are Open Source. So, we are looking for stakeholders, but also contribution back. And my third takeaway is that you don't have to use the whole library. You can cherry pick small thing. For example, I can just take the button and use it. I don't have to import the whole library. So, it makes it very easy to use small parts of it without depending on the whole the whole library.

Karsten: Okay, do you have one to add, Fabian?

Fabian: Fundamental will keep your performance in every browser on a very good level because it doesn't add a lot of JavaScript or a lot of CSS files because its component based, if you want to say so, and you just import the stuff you need and you don't need to include the whole library itself.

Karsten: Okay, because this is something people might be worried about, that they get a lot of overhead that they don't even need, right. Okay. So, it keeps it performant and it's cool. Then let me thank you for being here today, Fabian and Deno. It was nice to have you here and thanks to you all for listening to the Open Source Way. If you enjoyed this episode, please share it, and don't miss our next episode. Always published on the last Wednesday of the month. You'll find us on openSAP and our regular podcast distributions like Apple podcast, Spotify and the likes. Thanks again for listening and thanks again for being here. And let's all say goodbye. Bye bye.

Fabian: Goodbye. Thank you.

Deno: Thank you and bye.