2017-10-11 TSC

MedBiquitous Technical Steering Committee

October 11, 2017

Attending: Sascha, Jon, Joel, Scott Kroyer

1. Comparison of JSON API and OData for usage by MedBiq API specifications

Sascha noted that he wanted to follow the broader trends of our vendor community, but didn't have a sense of that. Joel noted that in our early standards work, we looked at what kind of technology our members had. Our smaller members did not have much in the way of middleware and such. For APIs, is that still the case? Are heavier hitters worrying about APIs?

Sascha replied that he hadn't seen huge change. They have limited resources and limitations on the complexity that can be thrown at their systems. They do see a demand for simple common REST APIs via tools like Tableau or Microsoft BI to draw data quickly. They want to draw JSON data without having to work in the web toolkit. Joel noted that for Curriculum API, hosters would be universities? Valerie noted the hosters would be tools, not universities. Scott noted they are not using either JSON API or ODATA for their implementation. Joel clarified they coded straight to REST. Scott clarified their existing implementation is not API based. the new one will be REST based. Haven't considered ODATA or JSOn API yet - just doing free form JSON. They are moving into standalone microserves that will need to interact and looking at ways to abstract them. using free form JSON approach will lead to problems, so they will start to look at more metadata in JSON. He leans away form ODATA, had bad performance with an experience last year. They will have to provide more in the data contract. Joel asked if he had problems with the tools or the spec. Scott commented the problems may have been mischaracterized. Possible that the performance problem related to an underlying problem.

Joel noted the other thing to consider is that our architecture allows for XML and JSON. Sounds like Ilios is pure JSON implementation. APIS transferring parts of MedBiquitous documents might be XML, new things might be JSON, others might be both. V noted xAPI does not use either. Joel commented most APIs don't use either. Complexity may be addressed through other infrastructure. but toolkits can make APIs easier and more consistent. Jon agreed many APIs stumble into these; it's way Ilios did. Other things they were using used JSON API. other groups may be doing the same. They use a javascript framework called ember, and that uses JSOn API to communicate to browser. New york Public Library and movietickets.com uses json api. Ruby on rails, too.

Joel noted that based on scenario gathering, is accredited, and answer may be yes or no, we want to be non-burdensome. But we also want to give direction to larger APIs with big aggregations where you need to navigate the structure behind the API. That's where JSON API is useful. D not like that JSON API created their own content types. It screws up people doing multipel content types. Jon noted you can send multiple content types with the request. It shouldn't interfere with XML. Joel noted the json api media type is required.Jon rpelied to get json api back, you need to send that content type. You can indicate yu want json xmo or jsn api, and allow content negotiation to handle. Scott note that for things that are eventually boolean responses, an API for that should be structured so that resource requested is certification. response is no response body, either 200 or 404. OK or not found. Jon agreed. Valerie noted there would likely be more data. Scott commented you could send more data, but not certified would be a 404.

Valerie sent a link to githib. Jon noted they bumped up against how to do filters, noting we should be consistent across specs. Joel asked how the filter would work. Would you specify one or more? Jon replied yes. Joel replied it is an "AND" ; Jon agreed. Joel noted the questions about arrays. Joel asked who the editor was. Sascha noted that with regards to trends, he asked Scott given what we are working with, could you foresee using ODATA or JSON API, making that work, and would one be more effective than the other? Scott replied yes and no. It might make interfacing with other APIs easier. he could foresee moving that direction. The likelihood of trouble free integration would be more likely. he may not extend that to the rest of their API. Sascha asked if he saw benefits or drawbacks to one or the other. Scott replied there is broader support for OData, but he does not have a seasoned opinion.

joel asked who the drivers of JSON API are. Jon replied it is an open source community effort. Dan Gephardt and Yehuda Katz developed it, both heavy hitters.

- Yehuda Katz
- Dan Gebhardt

Jon noted OData has more automated support. From his standpoint, it doesn't happen that way. Joel suggested meeting again - he will send some information out between now and the next call. Scott noted that OData is certified. JSON API is not.Joel noted there were multiple possible approaches. Scott added potential variability would not be that different than having XML schemas. Scott noted that using a consistent approach may have less benefits for their teams. the burden will be greater for very small teams. Sascha noted that explicitly calling metadata easily controlled with one that can;t be done in another. some thing might be more feaisble in OData than in JSON API. He leans to not constraining but giving guidance,

2. Possibilities for referencing one or more such specs from our API Architecture

3. Planning future activities, including discussion with the AAMC on federated identity