Ben Kutil

Ad Hoc, Healthcare.gov API Team

6/22/2018

**Moderator:** [00:00] Yep, that's fine. Perfect. All right, so to just kinda jump right in a couple of warm up questions, can you tell me about your role and responsibilities and kind of specifically as they relate to KPIs?

* [0:13] Sure. My role is acting as an Ad Hoc's product lead for our work on healthcare.gov in relation to APIs. Ad Hoc has responsibility for delivering and maintaining, um, for different kpis used in healthcare. I go, um, those apis are also made available or some of those apis are also made available to third parties, uh, typically, uh, either um, healthcare issuers or um, what, uh, the government calls a direct enrollment partners. Uh, these are private companies that work with the marketplace too. Okay. Uh, enroll people in healthcare plans that are available on the marketplace. Um, so as part of that, um, uh, in addition to those four Kpis, we maintain the, uh, authentication process and the ability to give, uh, an issue or a or a direct enrollment partner, a, you know, an api key, the ability to rotate that API key so that they have access, um, to pull that information out of the marketplace.

**Moderator:** [1:54] So it sounds like you kind of have a good sense of both user needs, like an end user needs and consumer needs as well as the developer side. And how do you decide what Apis to develop? Can you walk me through that process?

* [2:10] Sure. I think in general are the government partner we work with Center for Medicare and Medicaid Services, um, and the group within that really has an emphasis on building as much as possible all of their, um, tools api first and really with an emphasis on making data public. This is a organizational goal that they have, have, uh, as much as possible, um, making any of the data that they have available so that private companies can, can build on top of it. And um, yeah, not only offer their service, their government services, but also to create a, for lack of better word, a marketplace or additional competition for those services in the private market. So whenever there, whenever we are working with data that has, you know, that the organization has sort of outline is like they want to make public. Um, yeah, that's, that's one way that we started saying like, okay, well let's make an API to do that. Um, also, um, you know, we often have multiple tools or applications that make use of the same data and that's yeah, and another sort of opportunity or a signifier to say, hey, let's build this into its own service so that multiple tools can make a, you make use of it. So an example of that is marketplace Api, um, which is the, um, repository or database of all the, the health plans I'm using healthcare gov. that's also the database of all of the providers and drugs and facilities that those health plans cover. Um, and also, um, where some of the business logic around how uh, all of those things are, are costed out or calculated. Um, and all that information is maybe we use all that information within healthcare. I go across four or five different applications and so, you know, instead of each of those applications duplicating, you know, that information, it made more sense to build that as an api and allow them to. Yeah, each of those tools of it.

Moderator: [5:03] So it sounds like the availability of data is a major component of that process of deciding whether to build an API. How does, how does that relate to end user needs now? What, uh, what has been your experience with that?

* [5:21] Yeah, I think, in terms of consumer needs, it's essentially a transparent, whether we had this data available via an api or whether we had a tightly integrated application stack, a consumer would not, um, but not know the difference. Um, in thinking about, you know, the larger um, end user has sort of been defined for the marketplace. So again, like the issuers who are involved with third party partners who might be developed for them it was really building it as an API or having those apis available really does create those opportunities for then to build tools, uh, more easily prior to, again, using marketplace API as an example. Um, prior to that API being open for third party use, cms had put there plan information available to download as a flat file. Uh, so this was a two gigabyte file that you could download. Um, it only got updated, uh, quarterly if that, oftentimes it was only twice a year, um, and had none of the other rich information that was available within the marketplace around business logic or, um, you know, the providers, providers, facilities and drugs that those plants covered. Um, so while cms was making that information available for companies to use and companies were using it, um, for example, Health Sherpa, um, prior to using marketplace Api, uh, would ingest that big flat file every year and that was their source of data for, um, you know, for their tool.

Moderator: [7:45] Yeah. Yeah. You mentioned things like, of sounded like the flexibility is a major component as well. What has been your experience with rolling out, you know, api standards and then how do you balance that standardization and still account and allow for that kind of flexibility and innovation?

* [8:16] So a good example of where this has gone poorly is not an API that we develop, but it is an API that we have to integrate with the marketplace. A cms is in the process of really helping third party groups like Health sharper others, um, a takeover or have more of a say over the application process for healthcare as well. And so they've built a, um, an API for submitting applications, um, that API does some standards in terms of, uh, how a payload information is organized and the types of values then it's expecting, um, but it does not follow standards around versioning Api, which is often considered a best practice. That has severely hampered the development development team working on our own internal, uh, you know, the marketplace application, um, as well as the third party partners who are trying to build their own application process on top of this Api. And it has hampered them because, uh, every time this API has new changes that break existing functionality or change existing functionality, it breaks everyone's, uh, applications. Um, and so, um, it has really slowed down the flexibility or ability to innovate in that. Everyone now has to have a sort of a lock step deployment process to account for these changes that they're making. Um, you know, and typically, um, an API would follow a versioning best practice where, um, you know, there would be a version endpoint that was stable, uh, so that, um, the, you know, all the, uh, groups using that Api could develop against it and you know, sort of have their flexibility to do what they needed to do, um, but then also giving the team who's developing that API, the ability to develop the next version, um, independently of all of the groups using the API. Uh, so that when the time comes to make changes, uh, they'd be able to say, hey, we now have version two available, here's our schedule of, you know, when we see version one going away, you know, and please plan accordingly. And, and that's um, you know, that's one way to sort of allow for flexibility on both sides of the equation, you know, the development team working with the API and also the development teams working against the API

Moderator: [11:27] Makes Sense. What are other examples of best practices for Managing Api?

* [11:34] Sure. So there's a standard called the open API format, which represents a documentation standard among other things. Um, and one of the benefits of following that open format is that itself generates documentation, um, so that, um, you know, as you're developing an endpoint, um, which is, um, where you would send a request to get information from, um, you know, as the development team and as the, you know, the team in general that's maintaining that API instead of having to write up documentation around how to use that particular functionality or endpoint it is generated out of the code, you know, as it is being developed. Um, there's also many tools that can then use that open API format to generate testing tools. Uh, so an example of this is a, um, a pro product or tool or whatever come swagger, um, which can ingest that open API format and turn it into documentation as well as, um, a self running testing tool where a person could test each of the individual endpoints or functionality of that Api and see, you know, expected results. Um, so that's one example of a best practice in terms of a documentation. Um, there's a lot of different approaches for best practices in terms of how to format the, um, the request that someone would send to the API as well as the response that that API would give back. There is an example of that is called the JACEON Api Schema. Um, and essentially it's a way that, um, provides a, sort of a standard way for people to structure their responses and requests. Um, and I think the benefit there is it just reduces development complexity in that you always know, hey, this is how my request is going to come back and I've used it before because it is a standard and then, you know, things like versioning and a versioning deprecation schedule or know. Just other examples of um, best practices.

Moderator: [14:29] What's been your experience trying to get those standards? I guess let's, let's back that up a little bit. How ubiquitous is it for people to use these standards and if it isn't ubiquitous, how, what has been your experience, you know, integrating that into something like a government setting

* [14:46] Yeah, the group that we work with considers themselves at the forefront of like cms is API development, um, process or like a expertise, um, so they're very much on top of like adopting new standards and it has not been challenging. So I, I think in general that part has been relatively easy to say like, Hey, this is something we should do and here's why. And getting that adopted. I'm going back to that example of the application Api, um, or the applying api and then not following, um, a versioning best practices. Uh, there, it's been a challenge because it's not a group that we work with and it's not a group that are the, the, the um, our partners at cms have a lot of sway or control over, um, and this is just their way of doing things and they're like the gorilla in the room. And so everyone just like, sure, yeah, whatever. We'll see if that continues once, like, ah, there's less concerned from that group if their own internal teams have challenges. Uh, but I think once it is the third party private companies that are trying to use this thing and voicing their frustrations know there might be a little bit more pressure there to reevaluate their use of some of these best practices.

Moderator: [16:38] Right. So it seems like a kinda like a market influence to force them to adopt those standards. What are, what are some of those challenges that people experience when say for instance, if the standards are in the top 10?

* [16:53] Sure. I think, um, yeah, the, maybe the smallest one is like developer frustration of and like the impact of that frustration on the overall development timeline. So yeah, if we're not following, uh, if an API was not following a consistent way of structuring their payloads that you send in or that you get back, as a developer that's really frustrating and really impacts your development time. Same thing with documentation, you know, that's not only impacting development time, but that's also impacting adoption where, yeah, if I learned about a new API but there's no documentation or incomplete documentation or I can't easily, um, try out requests see responses. Um, you know, that's going to limit the amount of interest someone might have and in actually adopting and using that Api, um, and then you know, to things like versioning and uh, for example, with the, the API for applying, um, like that is going to fundamentally alter the development timelines of all the uh, applications or tools that are using that, uh, that api where they're going to have to coordinate releases. They are going to have to coordinate their development schedules to account for breaking changes and a lot of, um, you know, like those are the two big things that will impact, you know, not just individuals but entire organization as they're trying to develop against that API.

Moderator: [18:46] Kind of want to switch gears a little bit and move towards another topic. So we're interested in learning more about authentication frameworks. So when you're, when you're building an API or even from a consumer standpoint, how do you manage security and privacy concerns and then are there any special considerations you need to be aware of when you're working or creating with an api that manages authentication or authorization?

* [19:17] The one benefit for the API is that we maintain is that those apis are read only Apis, uh, so people aren't submitting there personal information into our systems which simplifies a little bit our security requirements and also the fact that all of our data is considered public, uh, also simplifies some of our application development security requirements, uh, in general, um, and that also like we have an Api key, uh, and the ability to set Api keys for sort of limiting access. But truthfully, that's not so much for security because all of our information is public. It's more for analytic purposes and the ability to maintain a specific traffic volumes, which sort of is a security thing.

Moderator: [20:31] Right. I'm just going to jump in and add a little color because I don't think so. What we're trying to do is basically like twofold. One, we need to allow veterans to like log in and manage who has access to their data. So do you guys have any third party, like I know there's like health navigators who can work on behalf of applicant's. Do you work on that product or not at all?

* [21:00] We do authenticate the call center reps who work on behalf of consumers, um, their information comes out of a database the same way, whether it's or it comes out of the API is the same way, whether it's a call center representative or consumer

Moderator: [21:26] like a private company that's acting on someone's behalf. How do you know that that person is allowed to see the person, the, like applicant's data.

* [21:38] So the, um, again from the applicant, from the applications point of view, like the actual code that's running the API requests are always the same. And so the um, authorization and authentication happens within the application itself to make sure that, hey, I can request based on my role and user that I'm able to request this information

Moderator: [22:19] So people become like credentialed in some way to request data on anyone's behalf?

* [22:26] Yes

Moderator: [22:26] Okay. So there's no framework now where somebody goes in and says this person is like my power of attorney or whatever?

* [22:30] no

Moderator: [22:34] Okay. Interesting.

* [22:36] . All consumer accounts are like a one to one relationship. And then all other support accounts are a one to many relationship. So the call center, there's a couple other, what we call like advance resolution roles and those all have access to any, uh, any consumer record that they're working on.

Moderator: [23:17] And then what about an individual, like a, like a parent, like somebody's applying for on behalf of their parents. Is that a use case that you guys.

* [23:32] Yup, but in that case the person who is doing the applying is considered the primary contact it is their accounts.

Moderator: [23:42] Got It.

* [24:44] Just in the application process they identify themselves as a non filer. So then the healthcare application is not for them. It's for the people that they're filing for.

Moderator: [23:59] So how do you manage, how do you check to make sure that that person is allowed to submit an application somebody else's path or is that a business process?

* [24:13] Not a part of the account creation process. And so we do LOA2.

Moderator: [24:22] Can you show me a staging? I see that in staging at it and at all. So I can see the flow.

* Sure

Moderator: Cool.

* [24;27] Actually I can send you a long powerpoint.

Moderator: [24:34] . I just want to see like, cause we have like VSOs who want to access and submit applications on veteran's behalf, so I'm just like curious about how other people make sure that that person is qualified is allowed to do that. Like delegated access.

* [25:11] Make a fake account real quick

Moderator: You don't have to do this now. I mean you can just send. We can talk about it later. I just want to see. Okay. I'll stop interrupting now. Yeah, no worries. I'm just as a heads up, we do have about three minutes left on our scheduled time. Is it okay if we go just a minute over then? Yeah, it'd be interesting to see that powerpoint later on. So how familiar are you then with like other identity proofing solutions like OAuth, or OpenID connect and how do you make a good user experience out of that?

* [25:52] We're looking at the authorization and authentication services on healthcare.gov, are currently an accustomed built solution. Um, we're looking to moving it to more of a, um, cots, you know, uh, something like OKTA, um, or you know, using OpenID connect to either work with OKTA or to work with CMS has a larger identity management tool could interact with um, and um, because right now the healthcare.gov authorization maintains its own database of users, but then I'll pull from multiple databases of users and there's a long term goal of at least unifying the healthcare.gov and Medicare.gov account process so that maybe not as relevant right now for ourselves, but let's say you are going to be eligible for Medicare. You're keeping your one account and being able to have a transition with you as you go from healthcare.gov as well as actually being able to store or a definitely your medicare.gov records because that's currently collected, but you know longer term in the future, possibly also any of your healthcare.gov, related medical activity.

Moderator: [27:52] So how many accounts can someone have at cms? Do you know?

* [27:57] A consumer would probably only a consumer would have two.

Moderator: [28:00] Just Healthcare.gov and Medicaid.gov. Medicare.

* [28:05] Yeah. My Medicare.gov.

Moderator: [28:08] so where did the Medicaid people, their state accounts, not Federal accounts.

* [28:14] Medicare is Medicaid. Medicaid, yeah. That's a state.

Moderator: [28:22] So we're trying to do this and that in the VA too, but people have like 12 different accounts.

* [28:29] Yeah. That is not as big of a challenge on our work.

Moderator: [28:36] So when you're talking about, so are you guys talking about buying an off the shelf identity proofing system and tying it in with OKTA plus like a login and authentication system that you're going to bolt together. What are you guys doing?

* [28:52] Uh, the marketplace already has, um, integrations with a Department of Homeland, the IRS and the Social Security Administration and so they've built their um, you know, as part of the original healthcare.gov development. Like, the identification process that was built and so, we're probably going to continue to use that.

Moderator: [29:26] So it was their security department of Homeland Security and who else?

* [29:30] IRS

Moderator: [29:32] So when somebody registers for an account, they can just get in, but once they start putting in their information, there are checks about

* [29:42] Prior to starting an application. Um, there is a remote identity proofing step that, uh, calls on those three services

Moderator: [29:56] Awesome. I want to see that you guys control the look and feel of that

* [30:11] Its like that format of, hey, you previously lived on the streets, please select it. Whatever.

Moderator: [30:24] Do you guys have multi-factor authentication too?

* [30:30] No

Moderator: [30:27] Oh, okay, cool. Are you guys looking at that now?

* [30:32] It is a low priority thing like as a part of our requirements and I want to learn more about why it's not one of ours because people only log into our service once a year. They already have a hard enough time remembering their password. That like adding that two factor authentication helps secure that whole year. But it's just like another layer of complexity and frustration when it comes to redoing their healthcare. If anything I would rather like move away from passwords totally. And like have it be a link that they authenticate with because they do only use it once.

Moderator: [31:23] I have a million more questions for you. But we're over time. Who else should we talk to you from working on either the product side or the Dev side?

* I'm trying to think. There's a sub contracting. They used to be called super brilliant. They changed their name. They to date have been working on much of the API management tooling and then a novel has been working. Has responsibility for the communication and user accounts. The best people to talk to right now about that.

Moderator: [32.41] What does super brilliance new name again?

* [32:47] Crowball

Moderator: [32:49] And are they good then?

* [32:54] Yes

Moderator: [32:56] Who’s your uh,

* [33:00] [James@superbrilliant.io](mailto:James@superbrilliant.io)

Moderator: [33:05] We're a little overtime. Is there anything else that you think we should know that we haven't talked about yet today?

* [33:16] Nope, I think you guys have a hard challenge ahead of you