2023.12.15 Meeting notes

Date
15 Dec 2023
Time: 15:00 (CET)

Meeting link https://tib-eu.webex.com/meet/georgy.litvinov

Attendees
- Georgy Litvinov
- Brian Lowe
- Mark Vanin
- Dragan Ivanovic
- Miloš Popović
- Ivan Mrsulja

Agenda
- Meeting notes
- Access control
  - Search endpoints
  - Endpoint/procedure isolation
  - N3 Template substitutions with array of values
    - Removal of deprecated array implementation from N3Template and SolrQuery operations.
    - Idea: rewrite with JsonContainerArray
  - Draft PR https://github.com/vivo-project/Vitro/pull/426
- Reports
- Meeting schedule

Meeting notes (transcribed automatically)

Georgy: That this way we will cover something. I'll make a meeting notes transcribed as a last time. So we this will be recorded and than transcribed, ok?

Ivan: yeah it's okay with me.

Georgy: I haven't done anything with entry templates so far and have you sent me some details how it doesn't work, did you?

Ivan: I don't recall I did. I just presented it to you and you said that you maybe know what is. yeah you told me that you maybe know what is the problem.

Georgy: I knew that.

Ivan: I forgot. I really forgot what it was about. I can try it later today or on Monday because I was working on another project, quite a lot of work for that. And today I'm working all day on Captcha feature toggle, adding some tests, moving logic, refactoring it. So it takes as little space as it possibly can in outside classes, outside the provider.

Georgy: I see. Ok. And than what we have else? Yeah once again nothing about the endpoint procedure isolation because I was busý mostly with extended search and the access control improvements for recent pull request.

And the only thing is related to this excess control and the search so maybe you know some time ago there was components for Solr for in Dynamic API and I Think...

No?

Ivan: No, I wasn't working on that.

Georgy: Ok

Ivan: Well, Uh, yeah, Uh, how long ago was it? Because I'm here for like, uh, I'm on the dynamic AP project for about six to seven months. So that's how much I was working here.

Georgy: Yeah. Think a year ago half.

Ivan: Yeah. More and more. I came to work with Dragan on few projects, like, uh, 11 months ago, but working on Vivo, it was like four dynamic API, like, uh, seven, eight months, maybe.
Georgy: So it wasn’t completed and we didn’t really use Solr components in real endpoints so we haven’t try that and now it seem like we should work on that especially for there was an issue I think that was!!!bugged regarding, not an issue but a feature request. It was discussed on Thursday about a hiding some data so hiding some profiles and if we want really to hide something we should not allow to the access to the same data that was stored in the search index. But we have more than one component more the one endpoint at the moment at excess search index so firsts that of course the our standart search and currently I’m working on search filters that would allow us to create default filters for each roles we have so we would be able to attach some roles to a filter so than role is asking for some information the filter will be applied any way to that role and that way we can hide some data in search.

But that’s about the main endpoint at we have. But we also use search index for things like auto completion in forms so we have this endpoints that wait for some json request or maybe not json but some form data request and they return the labels of some individuals and I think labels is could be.. could contain sense differnt information too. So we need to take care of that endpoints related to that problem. And also we have views search index to show the list of individuals or for example on the persons lists or organisations list and for example maybe you have seen there is an index of objects in VIVO and this index also showing you basically all what is in VIVO and there are could be some internal objects here.

Ivan: Yeah, right, but you want to maybe add like a hidden property to one of the objects or one of the indexes.

And is it a saving indexes into multiple, like different index schemas, like persons, and then the different is organization unit, or you just get one indexing unit and every entity goes into that.

Georgy: Yeah there is only one index with multiple fields so the main field is URI I suppose and all the rest of course labels but other than that the special fields that we can create for filters as usually is dynamic fields.

And that was introduce in extended search. So the idea is to use this additional filters to filter or the hole index so not allow users to with specific roles like public role for example maybe self editors to access this search index at all an filter index I mean. Right with every request but to do that we need to other we need to reuse basically the same aproach sorry reuse the same filters on each endpoint or have one generic endpoint that would take care about the security, about this filtering...

Ivan: Uh, the way I did that, I did like a similar work in the past and, uh, it was done using Elasticserearch, but I believe the Solar Elasticsearch are similar in that manner of fetching data.

And, uh, what I did is, uh, for every search instance, I had one service. I did it in Springboard, but yeah, it's analogous to this problem. Uh, I had like a search service that you would, uh, it's a genetic service that gives the type of units because I put different stuff like organizations.

It was also a great system organization, it’s a researchers and stuff like that. It was put into different indexes and I would send the, the, uh, index name and the class type that I want to map it in Java to that, uh, uh, to that service.

It was like a generic, the, the, uh, parentheses and, uh, there, all the filters would be like, I would send the query that I would create in the backend. I don't know if Solr works on the same way, but for Elasticsearch, you have to create a query, then send that query and run it on the Elasticsearch system.

And then the Elasticsearch system returns the search gets. So it was like a middle step, like a middleware for it. The query was sent to that service and then all the filters and all the required, like, checks were made there. Like if something was flagged as unable to be read by someone, that would be added to the query like one must condition would be added to the query and then it would be sent.

So it will always go through the same filter. And if you want to change that filter that is on most of the endpoints, you just have to change it in one place.

Georgy: And did you reuse the same endpoint? Hi Dragan!

Ivan: I reuse the same, I reuse the same service for all the, the, the, uh, queries that you would send there.

Georgy: I see, I see.

Ivan: Uh, so just make one, like, I don't know how I can tell you that I know how Solr really functions with queries and such, but if you can make a generic service that would act as a middle of air between your application and the solrs search engines, that would be great. Uh, because you would have like a one-class one responsibility principle, and it would be really easy then to maintain it. And if you would add one new endpoint or 10 new endpoints that would access the search, you would just have to add the logic for them.

You wouldn't have to copy the logic for the required filter fields that you would like to add. So it's a smart thing to do.

Georgy: Dragan, we are discussing maybe similar issue was discussed on Thursday and yesterday about the access control and how are we going now I'm today was working on this pull request to apply different filters to different roles to have filters search as also to hide sensetive data in search index but what I thought today and I had also meeting related to this similar topic is that there are more than one endpoint that use search. So we have search index used by the classifiers list on the pages and it's also reused by endpoints that provide auto complition information so you send type of some objects and !!!you will return you the labels. Yeah and ideally I think and we discuss right now ideally we should have some generic logic hopefully in Dynamic API for the search in that way that could be Dynamic API procedure that queries Solr and basically reuse that procedure with multiple endpoints in that case we will always be shure that the same filters that user defined applied to each query no matter what query it is.

Dragan: So I see. So all requests to the solar will go through that API.

Georgy: Yes and it will also allow us to decouple VIVO from search implementation because basically in that case we would be able by creating some procedure behind that queries not Solr but for example Elastic search or in case we don't have nor Solr nor Elastic search but we have super fasts graph behind it to use the same thing with a Sparcle. So in that case we would have options but yeah that was the idea of the that search in Dynamic API was year something ago and yeah...

Dragan: I'm just thinking about the performance. So basically, we were always fetching all information from the Solr and then checking which of the user was requesting the data as the access, as the right to access that and so on, which means that there will be on the Java side a lot of, it might be a lot of filtering. I mean, if the response is quite huge.

Georgy: I think it's it would be having implemented this filtering in Java in this extended search pull request I see that it will be at list the same as efficient as now but if we get some point hopefully I'll have time to the cashing of this Sparcle queries than it will even faster because yeah while I was tasting current extendet search I founded there was sometimes it was up to half second I think. This logic to filter everything and get information from the Solr. But if we are able to cash the Sparcle queries than it will be much faster.
Dragan: Yeah.

Georgy: Yeah so basically it's the logic is basically should be the same as we have right now in this endpoint but it should be written in Dynamic API than it could be easily modifiable in accessed because every time we write Sparcle queries in to the Java code. Yeah it's much more complicated to test it and much more complicated to modify that.

Dragan: Yeah Georgy I'm just thinking about the following. So there is the administrator, there is some editor for the institution and so on. And in those cases, there might be some complex, let's say, privileges rules, whether they can access some piece of information or not, depending on the institution to which they are responsible or something like that. And in that case, we need some filters, which will decide whether some piece of information from the store is accessible for that user or not.

And it's OK, even if it's not perfectly. So the best performance and so on, OK, we give our best to make people that and so on. But if we are thinking about the anonymized, I mean, the not to all the users were just coming to the we want to search for that.

And I think there might be a lot of those people who are looking for some piece of information inside this one. For them, it's just a question of whether some piece of information is published or not, what you're implementing at the moment, right?

So you're trying to implement whether some individual page is published, and maybe whether some journal article is published or not. I mean, catalog record, is it published or not?

I'm thinking, is there any sense in considering adding one field in the solar or in the search engine, saying it's published or not?

Georgy: So yeah tell you how it already so this for public role that was basically implemented in extended search. So now you can see the web site of SSH digital, right?

So you see I'm logged in as route user here and I see 38000 individuals. What if I logged out for example and do the same search I see only 62240. So that's was implemented just the way you described that.

So there is... but... let me log in...

Dragan: Okay, so there's additional field in the search engine, right? Whether it's publisher.

Georgy: Yes and there is not one additional fields or this is customisable. So for example here I have search configuration because here it was done in a box I can see that, modify that but for example let me find this filter text type maybe text, order, public and there is a data property... I sorry...It's not in a fields in the filter value. Sorry. One second... Cultural object type yeah and is default public value yeah so basically this filter value is use so I defined here

as I remember I defined here or maybe went by the second I'll check it once again. Architecture drawing... not assigned...

Dragan: Georgy, but just explain it to me. Is this something specific for this instance of winter or it's so it's not possible? It's not individual code base right at the moment.

Georgy: At now in Vitro code base so it was part of my extended search pull request so it's already works but it works only for the public roles. The way it works you have default public I think this is the property as you see here.

Yeah I think the cultural object was yeah it was the right thing. So because it is here this filter value because we have this filter value assigned to a filter and this filter value is applied so is basically checks if the type as I remember if the value in that filter in the field is this one. In that case we only values with that field are show on. And currently it's not hard coded to some specific field you can just create new fields for example show you so here we have different filters and different fields. So and if you want to create and apply to specific filter you just create this filter with a field for example this overview and this is multiple string value dynamic field and than you create filter value and say that is should be default public so than this filter no matter if user go here and define something user would always have filtered results here, filtered by what he is able to see. And what I'm doing today I haven't finished that but I in a testing phase basically for some reason yeah I build it wrong. I modified so now there is not this data property have true or false value but now there is a property that allows you to define the role. Not only for public but for example for self editors, for admins, for editors and for creators or if you have a custom role than for your custom role. So in that way it will be more flexible so you would be able to filter for every role different things.

Dragan: Okay, so it's implemented. So what you presented, it's important for the search that you could add one field and say, this is the filter. What is the public? What is not public? And so on. But it's not working for browsing. I mean, for opening the page.

So it's just for searching, right?

Georgy: Yes, yes exactly. The problem is even than I create that there are still be will be ways for the data to leak. And I would like to investigate how many endpoints we have because I know about endpoints to have auto compilation that the first thing, we have some endpoints I suppose or maybe just code that access this search engine inside our template engine that load some data from there and ideally with dynamic API we should have it's centralised or we could centralise search component and multiple public endpoints should go to this component go threw all the filtering, apply all the filtering for each request and do there specific stuff. Yeah and as you see some for example here the performance is ok.

So even if I not log in everywhere works ok. Although there is this predefined filter because Solr is very efficient with filtering this thing.

Dragan: Yes, it's important that it's on the side of the Solr. It's not in our Java code that we are.

Georgy: Yeah yeah So it's only matter of configuration so there is this data values I'll show you. So yeah... that's...

Dragan: There will be more complex cases, meaning that if you're an editor for some institution or something like that, right?

Georgy: Yes

Dragan: And for those complex cases, it can be only implemented in some Java filtering, or you think it's also possible to.
Georgy: Yeah I think that can be implemented but for our current state I mean for current state we either need to copy the approach that was use in this... how it's called...page search controller to other endpoints yeah. And yeah maybe I'll show you so basically that's what I'm preparing it's I'll still managed to make that work but that's basically the configuration so here you have filter value. So that's what I have so I have a filter value an I want to show that on the contenten so we have a type of contents and I would like to specify that this filter value applied for roles self editor, admin and public and that will be added to the pull request once it's ready. And I hope if I'll find some time today I'll do that today will see I'll just might be a little bit busy after the meeting. Ok. So do you have any anything that you would like to discuss about Dynamic API else today?

Dragan: No, but I would like to give that higher priority for v1.16. So hopefully after we publish the 1.15 soon, I think I would like at least that dynamic API is really merged.

Georgy: Yes hopefully hopefully. We have a lot to do I would say to make it how say safe so the procedures and endpoints would be isolated. Yeah there is some things to do because we without at yeah it could cause a long term issues for us I suppose. Yeah I would be happy to see that the finally merged.

Dragan: So do you think it's realistic that it's merged? Let's say that we have everything on the naming API 1.14 or however you call that branch in June. And then we need some additional time for testing and hopefully September, October to try to.

Georgy: We'll see I think it really depends on how much time we will all of us have because... for me for example I see that next half the first half of next year for me would be...it's already looks like I should do three things to once and yeah I really would like this ???? and maybe than I will be I'll have to invest !!!mode to simplify my work with Dynamic API. Will see. I really would like to be to happen but practise says that for example an last half year I didn't have a much time I was busy. Maybe once we finished the access control and extended search it would be a little bit it easy for me. And I will be on vacation from 25 to 7 of January so I suppose let's meet next time on 12 of January. Is it working day for you?

Dragan: It should be. Is it Friday right?

Georgy: Yeah. Friday, friday.

Dragan: Okay.

Georgy: Okay. So next time we will meeting 12. Mark I suppose is busy so we will postpone discussing report generator until than. And yeah no progress yet with entry templates and procedure isolations so far but yeah we discussed something so. I don't have anything else to discuss today.

Dragan: Okay, then we have half an hour more for for ending the week. Apologies for being late for today meeting. Actually. Yeah.

Georgy: Yeah that's fine. Everybody busy at the end of the year.

Dragan: Yeah.

Georgy: So thank you very much and have a nice rest of the week. And see you.

Dragan: Yeah, see you. Thank you very much.

Ivan: See you. Bye bye.