Theme: Shanghai library FOLIO project
Time: December 8, 2020 07:00pm (EST) / December 9, 2020 08:00am (GMT+8)

Attendees:
Vincent Bareau (Enterprise Architect, EBSCO)
Gang Zhou (Project manager, Shanghai library)
Sha Jiang (Technical Director, Jiatu)
Lucy Liu (Product Owner, Folio China)

Notes:

1. Is there any plan/discussion/proposal for moving historical data to historical tables?

Vince:

- The question can be interpreted in different ways: one is what we do with data after it reaches its retention time limit; and another one is whether we keep active historical statistical data within the platform as an ongoing basis.
- Do we keep active historical statistical data in the platform? We do not keep historical statistics on the platform. That's been assumed to be done by external applications or systems that would provide reporting and things like that. And they would manage to keep the historical tables as they pull data in every time they get your history.
- Do we have any policies in place for retention time? This is also not directly managed by the Folio project itself, but left to the hosting organization. The hosting organization can choose to host whichever way they want to implement their own retention policies. And they Implement whichever policies they want for archiving purposes essentially.

Sha Jiang: We are interested in statistical historical data so we can do reporting on this and can compare circulation from one to another.

Vince:

- External systems would be responsible for providing us with a reporting at this point. You collect the data. And the historical snapshots would be maintained by the external systems.
- But there is one exception. There is support for the circulation log. The circulation log is designed to maintain the history of circulation around items. That could be used to provide reporting within the system if needed. So we do have one place in folio where we are maintaining a log of historical events.

Gang Zhou: We have high-volume circulation every year, approximately 6 million transactions. Can we archive circulation statistics in an external system? Otherwise, the performance can be a big problem after one or two years.

Vince:

- Yes, I do think it's possible. You can implement the data retention policy that you want. And you can choose what you do for archiving after you implement data retention.

- I believe there's also an additional feature which is to erase circulation logs. Transaction logs for circulation have been identified as sensitive data. Some libraries want the ability to erase transaction data on a regular basis.

- You could Implement those three things together: to put a retention policy, to choose however you want to archive it, and then to make use of the erasing of the transaction log to clean up basically. This is something you have to come up on your own.

2. Progresses on modules for statistics

Lucy:

- Lucy provided the link to the wiki space (https://wiki.folio.org/display/DD/Circulation+Audit+logs) where conceptual model, high-level solution structure, Low-level design (Transient architecture), Open questions and Requirements can be found.

- Lucy reached out to @Steph Buck, PO in charge of circulation logs. Per what Stephanie said, the Circulation Log included in the Honeysuckle release was the latest development progress. Jira issue UXPROD-1703: Circulation log of all actions filtered by a patron and/or an item with additional filters.

Vince:

- The one area where we have completed some progress in terms of statistics is in the circulation log. So the circulation log is capturing specific events as part of circulation transactions and storing them in an independent recording mechanism. This is something that can be exploited for analysis and other types of reporting.

- Did you have other statistics that you were interested in besides circulation?

Sha Jiang: Inventory is also very important to us.

Vince: This is not addressed in folio. We have the expectation that reporting systems should sit outside folio, collecting data from folio and reporting statistics within that system.

Lucy: Is that LDP (Library Data Platform) that we discussed earlier in the community?

Vince:

- Yes. LDP is an example of the external reporting systems. LDP is a working progress. It is a community-run project that is designed to effectively collect data from folio and make the data available in a monolithic database through relational queries. So you can use SQL queries to analyze your data.

- It's not complete at this point. There are many things that are not included in their datasets. They rely on reload of the data every time. It doesn't have a synchronization

mechanism for keeping in sync with folio. They also support historical snapshots. Basically every time you do a load, it creates a snapshot of that. If you do another load, then it makes another snapshot. And you can potentially compare the two snapshots. It's expected to work a daily upload of data.

- I don't know if it is capable of running at the scale that Shanghai would be needing.
- Initially the project was looking at two different storage mechanisms for the data: one was the Postgre database and the other one was to use a professional-grade data warehouse like the Redshift service from AWS. At this point, the implementation support for Redshift has been discontinued. They're focusing only on Postgres. So whatever they have is currently limited to the capabilities of Postgres in terms of how they will expose the data and how the data is queried.

3. Should each microservice use its own PostgreSQL database or all the microservices use a single PostgreSQL database? Are there recommended best practices at other institutions or from the community?

Vince:

- According to microservice principles, each microservice is able to and is responsible for managing its own data. So each module can specify its data storage location. And they can all be different.
- But in practice, what happens is, for efficiency of operations, hosting organizations will in fact reuse the same database for all of the microservices. It doesn't matter at that point because the microservices are not going to be attempting to directly access other modules' storage. At the storage level, that would be a big problem. And it's one of the criteria that we look for in any review of storage code. So technically, they're all on the same database for the most part, but they don't know that. So individual modules don't know and cannot assume that other modules are storing in the same database.
- It is quite possible that in the near future, we may probably make use of the concept that different modules may choose to store their data in a different database. And that database could be located in a completely different service or location, somewhere else. It doesn't mean that every module will be using a separate database. It means that some modules will be using a separate database.

4. Currently item status is defined in the raml and can only be changed by developers. Is it possible to store item status in the postgresql, so system administrator could add other status?

Vince:

- If I understand question#4 correctly, you are referring to the definition of Status in item.json which contains an Enum.
- This is one of the problem areas in the Folio design and implementation. There are a number of bad decisions that are revealed there:

    - the principles of microservices have been broken by having other domains (circulation, acquisitions) maintain their state in the Inventory domain;

    - Some of the values represent a hard-coded lifecycle for an item and therefore cannot be deleted or modified. I view these as technical debts and it is my position that they need to be corrected. But as for now it is important not to remove the existing values because something will break. There is expected to be a status value as part of the workflow. Some piece of operations or transactions will attempt to set that value. You'll get an error if it's not there. And the workflow will break. Similarly, you might be able to add additional values if you want, but I don't think it would be useful because if you add your value, presumably that means that you would want to change the value and set it at some point to the value that you created. But that value is going to be overwritten by one of these workflows. If one of the workflows is run, it may just choose to change the value to the value that it wants because of circulation or whatever else.

    - This implementation also does not lend itself well to localization. Because it's hard coded in the json schema file, it's not easy to implement localization to translate these to Chinese. Maybe you're thinking you could create additional values or entities in order to replace the text in the labels with Chinese characters instead. But I suspect that might break things also. I don't know for sure. You have to try.

Gang Zhou: We will not change the default value. We want to add some other data for localization purposes. The only way is hard code in the json file, isn't it?

Vince: I don't know if it's the only way. But I think if you add some for localization reasons, it means you will want to use them. So in order to use them, that means you will have a different workflow. And that workflow will not be compatible with the workflow that is being used throughout other components of the folio project. I am afraid you might not be able to achieve the solution you're looking for if you're hoping to do it by adding values because the values will not be used. And if they are used, they break something else.

Sha Jiang: Is it possible that we add another field to the item object, for example, circulation states, physical states?

Vince:

- Technically it's possible. Then you have to figure out if you're doing it within the main circulation object, in the main branch, or you are making a local branch for your purpose.

- There are some discussions going on right now for the last few weeks. And they will be coming to a conclusion soon. We're going to present results to Tech leads and then maybe to the Technical Counsel around how we handle reference data as part of upgrades. So this means that when we do an upgrade, there are currently some issues with data collisions, not for item status, but for other fields.

- An example might be material types. It's easy to add definitions of material types. Then when you do an upgrade, there has to be handling of the upgrade system to

recognize that there have been values added to material types and these have to be compared and merged with any changes to material types that are being part of the upgrade.

- In order to handle the situation, we are proposing that we split the representation of the data into the default data and the operational data. Default data is defined as part of the release. This is the standard default data. Folio will take a copy of the default data and use it for running the system. Then customizations can be made on the operational data but never on the default data. So we're introducing a more flexible system for handling customizations of this sort of data. That would allow for better localization support. That's the goal.

- So far that's not something that is going to be applied to the specific problem of item status because that is a complicated and twisted knot. That's not good. We're going to have to address it. I will take your request for the desire to be able to better handle this and add this to the list of reasons why we need to fix item status because it is just not working properly right now.

- To get an idea of how it would be better, I gave you a link in that response to material type. Material type is handled differently. You'll see that instead of having a hard-coded list of values in an enum, material type refers to another construct, which is basically a bunch of reference data definitions. And those reference data definitions can be customized, can be added to, and can be localized. This is the model we want to do for the entire system. But we also have to remove dependencies on workflows and remove dependencies on external systems that are depending on items to maintain its status like circulation status.

  Lucy's note: Vince's message on Slack "A more desirable implementation can be seen with Material Types where values are loaded separately from Reference Data: https://github.com/folio-org/mod-inventory-storage/tree/master/reference-data/material-types."

- Can you make the system work the way it is now or is it a blocker to you?

Sha Jiang: When will the changes happen?

Vince:

- The proposal will be presented this week. Implementation is not scheduled. But I think implementation would be most likely in Iris release.

- But again even if that does get implemented, it does not solve the item problem because item status needs to be fixed before that solution can be applied to it.

- I will raise the issue. I will take your requirements as another reason why we need to fix this now and not later.

- Maybe you can report this problem in Jira so that I can reference it. If you have an idea of what exactly you want to add in another field, then I recommend that you put in the fact, and then how you would like it to work, or what your proposed solution is for an additional field.

- Potentially you could help fix it if they agree. The solution may be, like you said, to create another field. The example of how it would work better would be the material type. So if you know the kind of field you want to create and we introduce your field to inventory, we could do it in the same model as material type. So you could define a field. And it would be referencing externally defined reference data instead of hard-coded values.

Sha Jiang: We will create a Jira. (**To do**)

Lucy: Under which category would you suggest that we report this defect?

Vince: The inventory project probably. Unable to localize the status.

Gang Zhou: Do you have a timeline for the development?

Vince: No timeline. I'm hoping the development can happen in R1 release. You can do it and give back to the community. It would be implemented and released in the next release, especially if it is described as a bug, as a defect.

5. Attendees agreed to skip the meeting that was originally scheduled for December 22/23 due to the holiday break in the US. The next meeting will be January 5/6, 2021.