

Theme: Shanghai library FOLIO project

Time: June 21, 2022 07:00pm (EST) / June 22, 2022 07:00am (GMT+8)

Attendees:

Vincent Bateau (Enterprise Architect, EBSCO)

Gang Zhou (Project manager, Shanghai library)

Sha Jiang (Technical Director, Jiata)

Lucy Liu (Product Owner, Folio China)

Notes:

- 1. We did some actions in an app, then switched to another app by clicking the menu bar on top of the screen. We tried to return to the page where we had left by clicking the app in the menu bar again. However, we landed on the main page of that app. Is the navigation across apps designed to be like this? If yes, is there any plan to change the design? (Sha Jiang) Are there ways that the user can return to the last page that he viewed in an app after he jumped to another app? Can we make it configurable for users to decide where to land, the last page viewed or the main page? (Lucy)**

Vince: This is a question of whether apps can maintain state so that when you return to the app you are placed where you left off.

- State persistence is primarily being maintained through the URL Navigation capabilities (specifically the query parameters) which are automatically being kept in sync through stripes-connect.
 - stripes-connects maintains that navigation URL using the “redux store” for persistence
 - the “redux store” will also persist other values which are not part of the navigation URL
 - however those will be lost by a browser refresh (since that action will consume the navigation url only)
- There’s another level being managed through the localStorage - including sizing - which persisted at the browser level. Which makes sense since the sizing is a browser-level capability.

Not all apps implement this state persistence. If the app is implemented with stripes-connect, it would get the navigation URL capability with no extra work. However, it would have the opportunity to explicitly disable the capability if so desired.

This is not currently a user configurable option.

Sha Jiang: When a librarian needs to do some other work (such as circulation, orders, etc.) in the cataloging process, the librarian switches to other APPs and then switches back to cataloging, only to find the content just entered is gone. What is the solution?

Vince: This is an additional use case that a librarian may need to switch apps while in the middle of working on a catalog record (for example). When returning, the partial data

entry was not saved and will be lost. This is a different case of persistence, it is not about navigation but about data entry. The simple solution is to require saving the entered data before exiting the cataloging application. To assist this the UX could provide warning messages that data would be lost. However, there is the situation where work in progress cannot be saved because it is not complete (and would fail validation). For this, additional functionality would need to be added to the cataloging app to allow saving data in “draft” or “incomplete” form. The persistence of work in progress is a feature of the individual app.

2. There is the login api /bl-users/login, but we have not found the logout api. What is the logout mechanism in FOLIO? (Gang Zhou)

Vince: Folio does not currently provide a logout API. This is partly due to the fact that Folio does not maintain a session object. The “login” functionality is really “obtain a new access token”. Once the user obtains the token, access will be granted as long as the token is valid. There is a logout feature in the UI, but I am unsure exactly how it is implemented. I suspect that it is handled entirely in the UI code by eliminating the access token.

Gang Zhou: FOLIO is typically for internal use and can be considered as an Enterprise Application. I think the functionality “login & logout” are common features of enterprise applications.

- From the perspective of the internal security, the function logout is required.
- In circulation scenarios, multiple librarians often use the same Circulation workstation. If the previous librarian does not logout after leaving, the subsequent librarian will not remember logout and login, and their operations are indistinguishable.
- If a librarian login into FOLIO unintentionally on a public computer in the public area of the library, the URL will be kept in the history of the browser. Some patrons may click into FOLIO, and some mis-operations may affect FOLIO data.

Sha Jiang: How to enable tokens in the new release? Is it through the configuration file or through the interface call?

Vince:

- A solution is being worked on to introduce the use of refresh tokens in addition to the current access tokens. This is the typical solution to a stateless design (i.e. one without a session object). The server is not aware of which users/browsers have been granted access via an access token. The solution involves issuing a short lived access token along with a longer lived refresh token. When the access token expires, the refresh token is used to request a new access token. This request is the opportunity for the server to deny the new access token, and thus effectively implement a logout.

- At this time Folio has the ability to enforce access token expiration, but there is no mechanism to implement refresh tokens. Implementing refresh tokens will require updates to the UI modules. Therefore access tokens are set NOT to expire, Otherwise this would create a poor user experience and break some integration to external systems that currently require access tokens never to expire.