iConnect 200676 Design Approach - Ability to Poll Web Services

Project Overview

This enhancement will introduce the ability to poll web services in FireLight. FireLight is now able to call out to a web service at any defined interval to attempt to receive a response which allows the user to continue on within FireLight while the call out is running in the background.

When a call has been placed that requires polling, there will be a toast in data entry that is customizable in the provider communicating that they may continue with their application and the display status will be changed to "Pending Response" (a sub-status of Data Entry).

Once a response has been sent back to FireLight, there will be another toast in data entry that will be customizable in the provider communicating next steps. A message will be sent in the message center, which will allow an e-mail to be sent out if the agent has "Send Message Center Emails" checkbox selected and the display status will be changed to "Response Received" (a sub-status of Data Entry).

For example, the Milliman web service can take several days to process an order due to various factors, so this functionality will provide the ability to poll this web service until a response can be retrieved.

Requirements / User Stories

Update app data real-time when a response is received from a polled web service

If a response is received back from a web service that is being polled and the user is in the activity, the data needs to be sent into the activity and updated real time.

The provider needs to get data asynchronously and then update the app data, including the apps session data in memory. The provider would call into FireLight Base to provide data items that need to be updated. This data can get updated in the main application data blob in case there is not an open session. In case there is an open session it would also be stored in a new DB table.

When the users update data and call the Value Changed action FireLight will check for any Async data that also needs to be included in the response. Rules will run normally and the appropriate data will be returned to the UI.

Acceptance Criteria

- Update activity data in real-time when a response is received from a polled web service.
- 3 part of message -
- UI will see a Popup message and a notification message in Message Center and a notification message in the email.
- DB Response field is updated with Complete.

Update display status during process

If one or more web services are being polled within an activity, set the display status of that activity to "Pending Response". The user would not be able to continue to signatures if there is a web service that is being polled.

If all web services being polled have received a response back, set the display status of that activity to "Response Received".

Acceptance Criteria

- Display status is updated to "Pending Response" when a web service is being polled.
- User is unable to move to signatures until web service response has been received.
- Display status is updated to "Response Received" when response has been received by web service(s).

Send message center notification when polled web service response is received

Once a response has been received for a web service that is being polled, send a message center notification to the user (who should also receive an e-mail if they have the appropriate option selected in My Preferences) that a response has been received.

Ability to insert message received back from the web service/provider to be part of the message sent in the message center notification.

This message will be toggleable via a property in the new rule node specific to polling web services.

Acceptance Criteria

- When a response is received from a polled web service, send out message center notification to the user.
- Insert data received back from the web service to be part of the message sent in the message center notification.
- Can control whether to use Message Center in the rule node that initiates polling.

Modify Support Tool

Modify the current Support tool to not interfere with the Workflow views. The new workflow items needed for polling should be hidden by default but will need to be viewable via an option. This is to avoid any unnecessary work or confusion to the support team.

Acceptance Criteria

- In tool, workflow entries with this new workflow type will not be visible by default
- There will be a button to view entries with this workflow type

Implement the ability to poll web services in FireLight via the provider

Implement the ability to poll web services at defined intervals within the rules to attempt to receive a response back. While this polling occurs, the user should be able to continue working in FireLight. The polling should continue regardless of if the user is still in the case that the polling was initiated in.

Add polling web service rule node with the following properties:

- How often the action is retried (Specify in minutes) (0 or -1 for no polling)
- Action Name
- Timeout period (i.e. 30 minutes, 2 days)
- Show/hide a notification message to the user when it is initiated within data entry
- Notify the user in the message center
- Message Center Message (Can be overridden by provider)

This should be available in both legacy UI and new UI.

[

Sample Provider Code to complete an async call

```
if (Context.ActionName == "TestAsyncCall")
        {
           Dictionary<string, DataItem> item = new Dictionary<string, DataItem>();
           item.Add("Test Data Item", new DataItem() { DataItemId = "Jacob Test DataItem", Value =
"This is a test value." });

           DateTime currDate = DateTime.Now;
           if (currDate.Minute % 3 == 0)
           {
              IApplicationService svc = Context.GetService<IApplicationService>();
              svc.CompleteAsyncJob(appID, new AsyncJobResponse()
              {
                 ActionName = Context.ActionName,
                 Message = "This is a test message from FSEB provider.",
                 DataItems = item
              });
           }
        }
```

Sample Provider Code to cancel an async call

```
if (Context.ActionName == "TestAsyncCall")
        {
Dictionary<string, DataItem> item = new Dictionary<string, DataItem>();
        item.Add("Test entry", new DataItem() { DataItemId = "Jacob Test DataItem", Value = "This is a test value." });

        IApplicationService svc = Context.GetService<IApplicationService>();
        svc.CancelAsyncJob(appID, new AsyncJobResponse()
        {
           ActionName = Context.ActionName,
           Message = "This is a test message from the provider.",
           DataItems = item
        });
   }
]
```

Sample Rule to start process:

```
<block>
        <callasyncaction name="TestAsyncCall" frequency="4" expires="15" sendnotification=”true” message=”Default Message”

 />
</block>
```

'frequency' and 'expires' are in minutes. This rule node is additionally explained in the Rule node help.

Acceptance Criteria

- Web service can be called (polled) intermittently at defined times in the provider
- Polling continues as the user works in FireLight - inside or outside of the case that the call was initiated.
- Polling rule node initiates the polling of a web service as expected.
- Frequency rule node property polls the web service at the value set in the property
- Timeout rule node property has the web service timeout at the defined time set in the property.
- Show/hide notification rule node property shows or hides notification in data entry.
- Toggle message center notification rule node property turns on or off the notification as expected.
- Message center message rule node property shows the message set in the property unless overridden by the provider in which case it displays provider message.

The Failed Submission Report should not be affected by Polling tasks

Since these long-running polling tasks will use the Workflow, they cannot be included in any reports that look at the Workflow data. The Polling jobs must be filtered out. The Failed Submission Report is the only report that currently queries the Workflow data.

Acceptance Criteria

- The Failed Submission report must not include polling jobs