Re-Open Problem Date Validation

Lab 6.2   10-15 minutes

Lab objectives

You will perform the following in this lab:

- Update a UI Action using GlideDateTime methods to verify that a Problem hasn't been closed for more than 30 days if a user attempts to re-open the record.
- Complete a Cloud Dimensions' Phase II requirement.

Business Problem

After analyzing the data, the Problem Management team determined that it is common practice for IT Analysts to re-open Problem records to document ongoing work performed on the same Configuration Item (CI). In some cases, the re-open was valid. In other cases, the re-open was not valid, as it was a new issue altogether.

After digging a little deeper, the team also determined where there was a valid re-open, the failure of the permanent fix presented itself in less than one week.

To eliminate the practice of incorrectly re-opening Problem records, the Problem Management team has put a policy in place stating that Problem records cannot be re-opened after 30 days. The team feels that this window is sufficient to catch any valid re-opens. They are requesting ServiceNow be configured to help enforce this policy.

Project Requirement

Prevent Problem records from re-opening if closed for more than 30 days.

A. Update a UI Action

1. Navigate to **Problem > All** on the Application Navigator.
2. Select a **Problem**.
3. On the form's Context menu (☰), select **Configure > UI Actions**.
4. Search the **Name** column for the UI Action **Re-Analyze**.
5. Select **Re-Analyze**.

> **TIP FROM THE FIELD:**
> *There is a quick way to open a specific UI Action. Navigate to the form. **Right-click** the UI Action above the Related Links. Select **Edit UI Action**.*

6. Examine the pseudo-code for the script you will write:
   - Read today's date into a variable.
   - Read the Closed date into a variable.
   - Determine the difference between today and the closed date.
   - If the number of days between today and the Closed date is > 30
     - Add an error message to the form page.
   - else
     - Allow the problem to be re-analyzed.
7. Write the script.
   a) Scroll down to the **Script** field.
   b) Enter the highlighted script:

```
1    action.setRedirectURL(current);
2
3 ▾  try {
4
5        var today = new GlideDateTime();
6        var closed = new GlideDateTime(current.closed_at);
7
8 ▾      if (closed) {
9            var dateDiff = GlideDateTime.subtract(closed, today);
10           var dateDiffNum = dateDiff.getDayPart();
11
12 ▾         if (dateDiffNum > 30) {
13               gs.addErrorMessage("A problem cannot be reopened after it
     has been closed for more than 30 days. Please open a new Problem.");
14 ▾         } else {
15               new ProblemStateUtils().onReAnalyze(current);
16           }
17       }
18
19 ▾  } catch (err) {
20       gs.error("A runtime error occurred: " + err);
21  }
```

8. Select **Update**.

B.   Test Your Work

1. Navigate to **Problem > All** on the Application Navigator.

2. Open any Problem record with a State of **Closed** and a Resolution code of **Risk Accepted** with a Closed date **before 30 days relative** to the current date.

   > **Note:** *You may need to configure your list view to match the example below.*

3. Select the UI Action **Re-Analyze**.

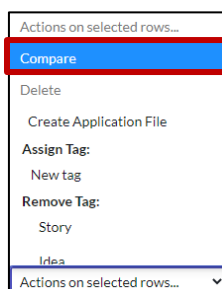4. Do you see an error message at the top of the form? If not, debug and re-test.
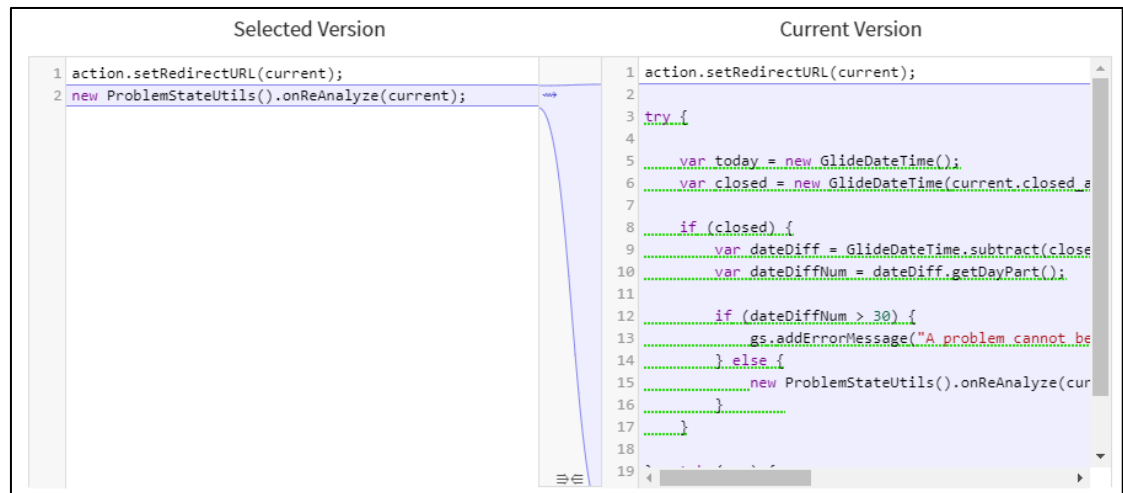

C.   Revert to the Original Version

1. Navigate to the UI Action **Re-Analyze**.

2. Scroll down to the **Versions** Related Lists.
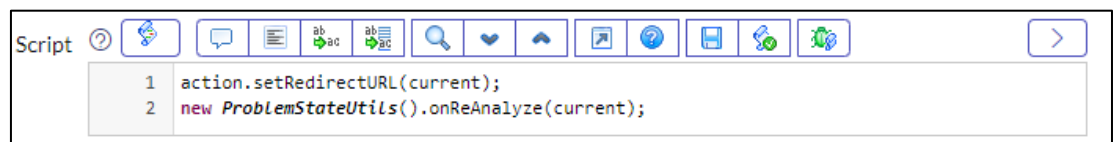
3. Select the **Current** and **Previous** versions.

4. Select **Compare** from the **Actions on selected rows…** menu.

5. Select the **Script** field.

6. Compare the **Selected Version** with the **Current Version**.

7. If the selected version matches the original script, select **Cancel** to return to the page Compare to Current. Otherwise, select **OK**.

8. At the bottom of the page, select **Revert to Selected Version**.

9. Select **OK** and a message appears to state that the UI action has been updated to the previous version.

10. Navigate to the UI Action **Re-Analyze** to confirm it has been restored to the original version.

Lab Completion

Well done! You have successfully completed the requirement and practiced comparing dates using GlideDateTime methods. Time to update your resume with these new skills.