ServiceNow Application Developer

Server-side Scripting > Exercise: Create a Business Rule

In this exercise you will write and test a Business Rule to prevent users from setting the *When needed* field value to a date in the past when creating *NeedIt* requests.

Create the Business Rule

1. If the *NeedIt* application is not open in Studio from the last exercise, open it now.
   a. In the main ServiceNow browser window use the Application Navigator to open **System Applications > Studio**.
   b. In the *Select Application* dialog, click the **NeedIt** application.

2. Create a Business Rule.
   a. In Studio, click the **Create Application File** link.
   b. In the *Filter...* field enter the text **Business** OR select **Server Development** from the categories in the left hand pane.
   c. Select **Business Rule** in the middle pane as the file type then select the **Create** button.

3. Configure the Business Rule:

    Name: **NeedIt When needed field date**

    Table: **NeedIt [x_<your_company_code>_needit_needit]**

    Active: **Selected (checked)**

    Advanced: **Selected (checked)**

4. Switch to the **When to run section** and continue configuring the Business Rule:

    When: **Before**

    Insert: **Selected (checked)**

5. Click the **Submit** button.

Write the Business Rule Script

1. Switch to the **Advanced** section.

2. Copy this script and paste it into the *executeRule* function in the *Script* field. Do not overwrite the template; paste the script after the *Add your code here* comment.

```
// rightnow stores the current time
    var rightnow = new GlideDateTime();
    // Create a GlideDateTime object for the When needed date
    var whenNeeded = new GlideDateTime(current.u_when_needed);

    // If the When needed date is before rightnow, do not write the record to
the database
    // Output an error message to the screen
    if(whenNeeded.before(rightnow)){
        gs.addErrorMessage("When needed date cannot be in the past.  Your
request has not been saved to the database.");
        current.setAbortAction(true);
    }
```

3. Click the **Update** button.

▼ **QUESTION**: What does the *current.setAbortAction(true);* statement do?

ANSWER: The *setAbortAction()* method is part of the *GlideRecord* API. Pass *true* to the method to abort the next database operation for a record. When you test the script you will see that ServiceNow writes an error message to the form when the database operation is aborted.

Test the Business Rule

1. In the main ServiceNow browser window (not Studio) use the Application Navigator to open **NeedIt > Create New**.

2. Set the *When needed* field value to a value in the past.

3. Click the **Additional actions** menu (☰) and select the **Save** menu item. Do not click the *Submit* button because that will take you away from the form.

4. Examine the error messages.

    ▼ **QUESTION**: You should see two error messages. One of the error messages was written by the Business Rule. Where did the second error message come from?

    ANSWER: The *Invalid insert* error message was written by ServiceNow when the database operation was aborted by the Business Rule script.

5. Create a *NeedIt* record and set the *When needed* date in the future.

6. Save the record. The record should be written to the database with no error messages. If the record does not save, debug and re-test.

Challenge

The *NeedIt* application is used to request goods and services from various departments. The departments need a little time to fulfill requests so same-day requests are not allowed. Modify the Business Rule script to prevent users from submitting requests for today. You will find the **scoped GlideDateTime API documentation (https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=r_ScopedGlideDateTimeGlideDateTime)** on the developer site to be useful.

▼ **CHALLENGE SOLUTION**:

There are many possible solutions. One solution is:

```
// rightnow stores the current time
    var rightnow = new GlideDateTime();
    // Create a GlideDateTime object for the When needed date
    var whenNeeded = new GlideDateTime(current.u_when_needed);

    // If the When needed date is before rightnow, do not write the record
to the database
    // Output an error message to the screen
    if(whenNeeded.before(rightnow)){
        gs.addErrorMessage("When needed date cannot be in the past.  Your
request has not been saved to the database.");
        current.setAbortAction(true);
    }
    // Challenge:  Do not allow same-day requests
    // Get the date portion of rightnow and whenNeeded (no timestamp)
    var today = rightnow.getLocalDate();
    var istoday = whenNeeded.getLocalDate();
    // Compare today and istoday to see if they are the same day
    if(today.compareTo(istoday) == 0){
        gs.addErrorMessage("You cannot submit NeedIt requests for today.");
        current.setAbortAction(true);
    }
```