Building Action Packages for Automation 360: Step List

Introduction to the Course

This course outlines the steps to build custom actions package and deploy them to Automation 360. Automation 360 pre-installs a set of popular packages, available in the Actions panel when creating a bot. Along with the existing commands packages, Automation 360 also allows users to create packages that contain custom actions and then upload them to the Control Room for reusability.

Introduction

This demo will help you build a custom package that contains an action that calculates thedate difference between two dates provided as input for the given use case.

Objective

In this demo, you will learn how to build commands packages, deploy them to Automation 360 and use the action in a bot.

Prerequisites

You need:

- Java Development Kit (JDK) 2.0.6
- Java IDE, it can be either
  - Eclipse, or
  - IntelliJ
- Latest Gradle plug-in in the IDE
- Install and connect the Automation Anywhere Bot Agent to an Enterprise Control Roomwhere you deploy the package

1. Building and Deploying Commands Package on Automation 360

a. Building the Java Project
   i. Visit the Automation 360 Package Development Kit Release Notes webpage.
   ii. Click and download the latest package development kit.
   iii. Rename the file with an appropriate name.
   iv. Extract the contents of the Package Development Kit.
   v. Specify the project name settings.gradle in the project root folder
      v.1. Open the settings.gradle file in any text editor such as Notepad.
      v.2. Change the root project name.
      v.3. Note: The project name should be a single word and cannot contain spaces.
      v.4. Name the project as DateCalculations.
      v.5. Save the changes and close the file.
   vi. Set up the Java Project and import the project as a gradle project
      Note: *You will need to follow the steps specific to your IDE.*
   vii. Edit build.gradle on eclipse IDE
      vii.1. Open the build.gradle file.
      vii.2. Provide the author tag value as **IT Bricks.**
      vii.3. Save the changes.
   viii. Edit **package.template** in the src\main\resources folder
      viii.1. Click the **package.template** file.
      viii.2. Change the action name.
      viii.3. Save the changes.
   ix. Create a Java file with the code for your usecase and add it to the src\main\java\com\automationanywhere\botcommand\samples\commands\basic folder.
   x. Delete the remaining sample code files.
   xi. Back in the Eclipse workspace, refresh the files.
   xii. Define the labels and descriptions for the Package and Commands in src\main\resources\locales\en_US.json file.
   xiii. Build the Java Project
      xiii.1. Open a command window.
      xiii.2. Go to the project root folder where the gradlew.bat file is located.
      xiii.3. Enter the command gradlew.bat clean build shadowJar.
      xiii.4. Run the command.

b. Deploying Commands Packages on Automation 360
   i. Login to the Automation 360 Control Room.
   ii. Navigate to the Packages page and click the **Add package** icon.
   iii. Browse and select the newly created .jar package.
   iv. Click **Upload package**.
   v. Click the **Accept, enable and set as default** button.

Congratulations! With this, you have successfully built the Commands Package.

2. Automating Date Difference Calculation Using the Newly Created Commands Package

a. Open the Excel sheet from the desired location on the device.
   i. From the Actions panel, drag and drop the **Excel advanced: Open** action to the bot editor.
   ii. In the Action details panel, click **Desktop file** and enter the file path with name.
   Note: *In all such instances, you will have the option to either enter the file path manually or browse and select the required file. You can choose either of the two.*
   iii. Open the file in a read-write mode and select the **Sheet contains a header** checkbox.
   iv. Click **Save**.

b. Create variables.
   i. Create the following variables
      i.1. sdate of type Datetime
      i.2. edate of type Datetime
      i.3. count of type Number
      i.4. diffindate of type Number
      i.5. strcount of type String
      Note*: Variable names shown here are examples. You can give any name for the variables.*

c. Loop through each row in the Excel sheet.
   i. From the Actions panel, drag and drop the **Loop** action into the bot editor.
   ii. In the Iterator list, select **For each row in** worksheet from the Excel advanced section.
   iii. In the **Loop through** field, select the **All rows** option from the drop-down list.
   iv. In the **Assign the current value to this variable** field, create and assign the variable to the output.
   Note: *Make sure that the variable is created with the variable type as* ***Record****.*
   v. Click **Save**.

d. Convert the string variable to a Datetime variable.
   i. From the Actions panel, drag and drop the **Datetime: Assign** action within the loop.
   ii. In the **Select the source date time variable/value** section:
      ii.1. Enter **$record[5]$** for the **Enter the date time** field
      ii.2. Select **ISO_LOCAL_DATE ('2011-12-03')** In the **Prebuilt formats** drop-down list.
      ii.3. Select the **sdate** variable from the **Select the destination datetime variable** drop-down list.
   iii. Click **Save**.
   iv. From the Actions panel, drag and drop the **Datetime: Assign** action within the loop.
   v. In the **Select the source date time variable/value** section:
      v.1. Enter **$record[7]$** for the **Enter the date time** field
      v.2. Select **ISO_LOCAL_DATE ('2011-12-03')** In the **Prebuilt formats** drop-down list.
   vi. Click **Save**.
   vii. Similarly, convert the **Cancel Date** string variable to a Datetime variable.

e. Calculate the date difference between the Booking Date and the Cancellation Date.
   i. From the Actions panel, drag and drop the **Date Calculations: Date Difference** action within the loop.
   ii. In the **Start Date** field, assign the **sdate** Datetime variable.
   iii. In the **End Date** field, assign the **edate** Datetime variable.
   iv. In the **Assign the output to a variable** field, select the **diffindate** variable.

f. Convert the stored number and the count variable to string variable.
   i. From the Actions panel, drag and drop the **Number: To string** action within the loop.
   ii. In the **Enter a number** field, assign the **diffindate** variable.
   iii. In the **Assign the output to variable** field, create and assign the variable to the output.
   Note: *Make sure that the variable is created with the variable type as string.*
   iv. Click **Save**.
   v. Similarly, change the count variable to a string count variable.

g. Set Date Difference value to Cancel Window in the Excel worksheet.
   i. From the Actions panel, drag and drop the **Excel advanced: Set cell** action within the loop.
   ii. In the Cell option section, select the **Specific cell** and specify the string count variable along with the Column heard for the Cancel Window column in this format **I$<stringvariablename>$**, where I is the column header.
   iii. In the **Cell Value** field, assign the output of the string variable that holds the converted date difference value.
   iv. Click **Save**.

h. Increment the counter to move to the next row of the Excel sheet.
   i. Drag and drop the **Number: Increment** action within the loop.
   ii. In the **Enter number** field, assign the count variable to it.
   iii. In the **Enter increment value** field, increment the value by 1.
   iv. In the **Assign the output to variable** field, assign the output to the same count variable.
   v. Click **Save.**

i. Close the Excel file.
   i. From the Actions panel, drag and drop the **Excel advanced: Close** action.
   ii. Check the **Save changes when closing file** checkbox.
   iii. Click **Save**.

Congratulations! With this, you have successfully created the bot that can calculate and add the respective difference values in the Cancel Window column.