Managing Queues in Studio

[**SUGGEST EDITS**](https://orchestrator.uipath.com/docs/managing-queues-in-studio)

**Adding Items to a Queue**

1. In the **Queues** page, create a queue.
2. In Studio, create an automation project that, for example, collects certain figures from an Excel spreadsheet and stores them in a variable.
3. From the **Activities** panel, drag an **Add Queue Item** activity to the **Designer** panel.
4. In the **Properties** panel, in the **QueueName** field, type the name of the queue created at step 1, between quotation marks.

**Note:**

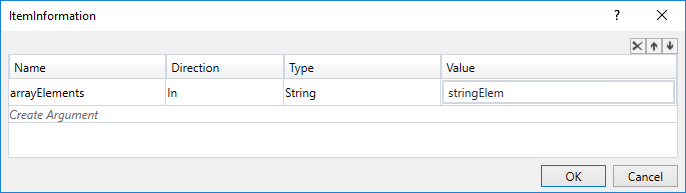
The queue names are not case sensitive. For example, "Test\_QuEuE" and “test\_queue” are the same.

1. From the **Priority** list, select **Low**, **Normal** or **High**, depending on the importance of the items that are added by this activity and how fast you want them to be processed.
2. (**Optionally**) In the **Reference** field, enter a string or string variable to act as an identifier for a queue item or a group of them. (The reference can be used to link your transactions to other applications used within an automation project. Additionally, this feature enables you to search for certain transactions, in Orchestrator, according to the provided custom reference.)

**Important!**

References are not compatible with the 2016.2 or lower versions of Orchestrator or Robot.

1. (**Optionally**) In the **Deadline** field, add a date until which the items must be processed.
2. (**Optionally**) In the **Postpone** field, add a date when to start processing the queue items.
3. Next to the **ItemInformation** field, click the **Browse** browse button. The **ItemInformation**window is displayed, that enables you to add data to each created queue item.

[](https://files.readme.io/1f69575-queue_item_information.png)

**Note:**

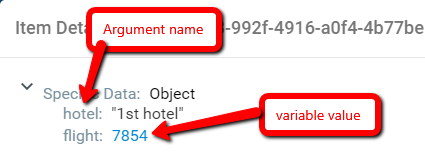
The value of the string arguments added in the **ItemInformation** window cannot contain the following characters: [ and " ".

1. Click **Create Argument**. A new blank argument is created.
2. Change the name of the argument to something more suggestive for your needs.
3. Change the **Type** of the argument to what you need. For example, if you want to store numerical values from transactions, select **Int32**.
4. In the **Value** field, enter the value that you want to add to queue items or a variable that you use to store a specific value.
5. Click **OK**. Your settings are saved.

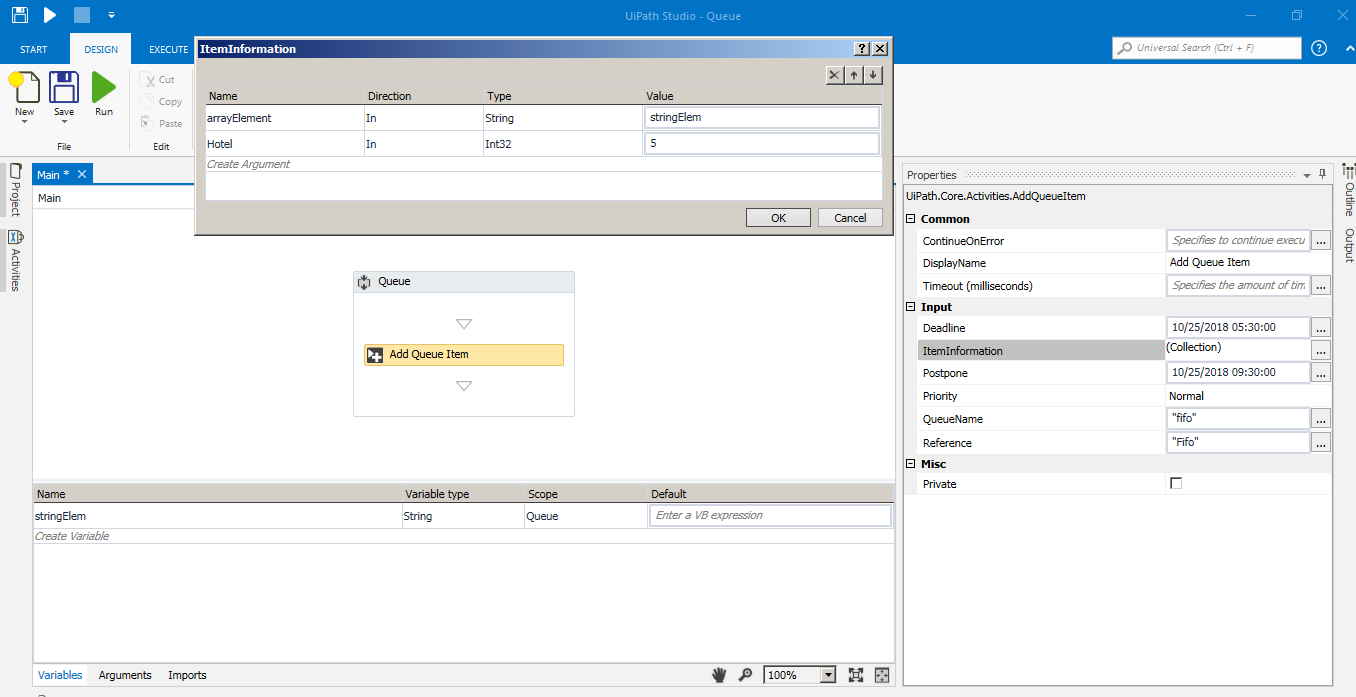
**Important!**

Place these types of automation projects in a loop. Otherwise, only one item is created and populated in Orchestrator.

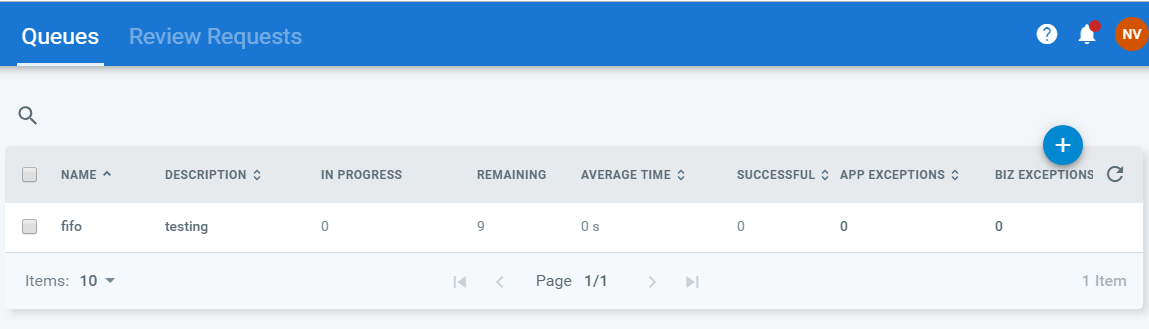
The data added in the **ItemInformation** window in Studio is displayed in the **Item Details**window, under **Specific Data**, in Orchestrator. All items added to queues have the status set to **New**.

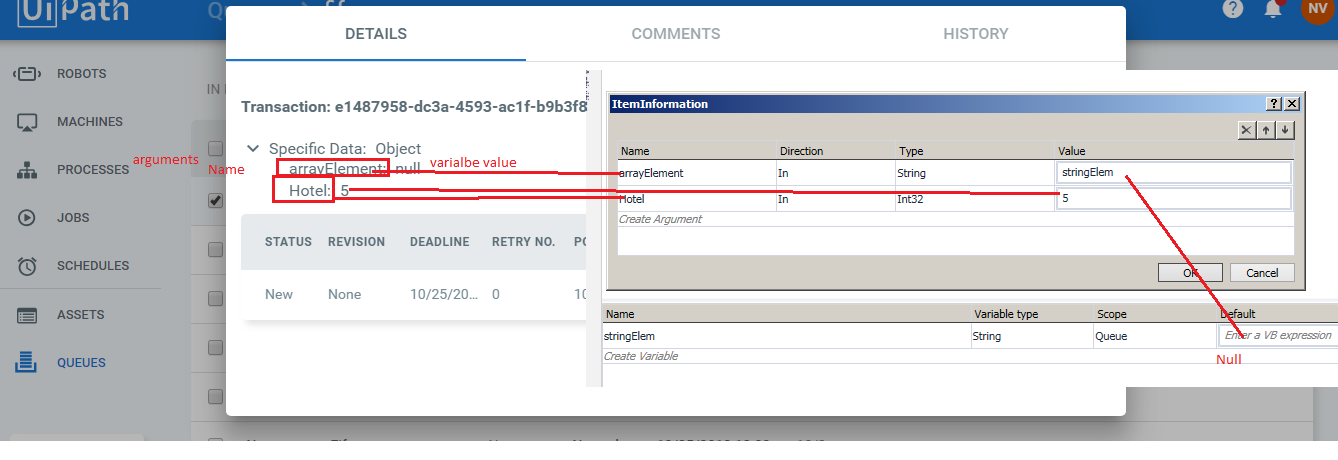
[](https://files.readme.io/cc6e0c6-item_details_window_specific_data.png)

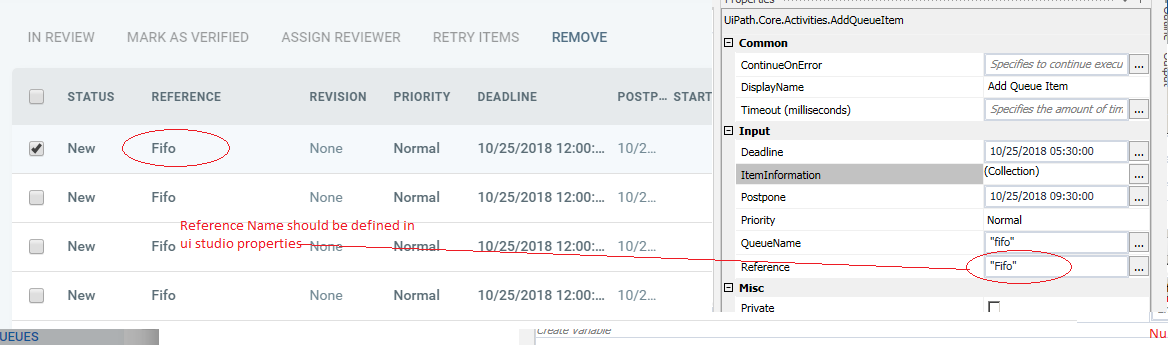
Screen shots

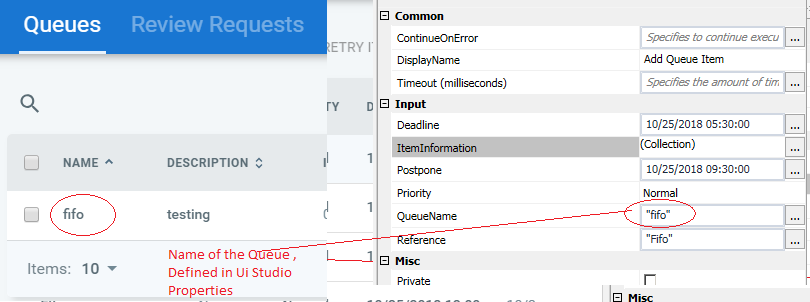


**After publishing from Ui Studio -> navigate to Orchestrator -> use the update package (we can update the package by using ‘use’ option in view process in process Tab**



****

****

****

**Sample code of the above can be find below**

****

# Getting Items from a Queue and Changing their Status

1. In Orchestrator, consider you have a queue populated with items.
2. In Studio, from the **Activities** panel, drag a **Get Transaction Item** activity to the **Designer**panel.
3. In the **Properties** panel, in the **QueueName** field, type the name of the populated queue from Orchestrator, between quotation marks.

### Note:

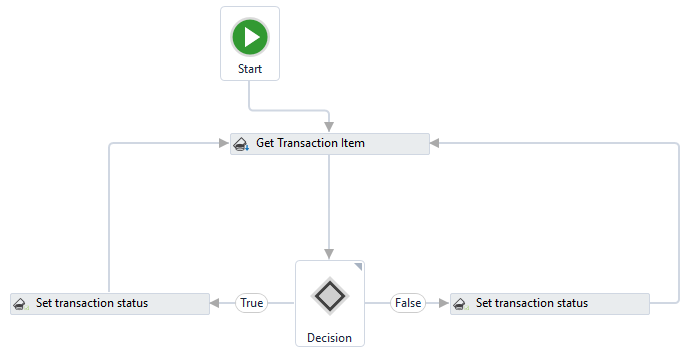
The queue names are not case sensitive. For example, "New\_QueUe" and “new\_queue” are the same.

1. Create a QueueItem variable. This variable stores the first item from a queue, according to its priority and deadline. When placed in a loop, the Robot iterates through all queue items. After a queue item passes through this activity it is NOT processed, and the status of the item changes to **In Progress**.
2. In the **TransactionItem** field, enter the QueueItem variable created previously.
3. From the **Activities** panel, drag a **Decision** activity under the **Get Transaction Item**activity.
4. Create your automation requirements. For example, add the transaction items to an Excel spreadsheet and send e-mails with this information.
5. In the **Properties** panel, in the **Condition** field, type a condition (for example, the queue item is empty) based on which the status of the item can change.
6. From the **Activities** panel, drag a **Set Transaction Status** activity and connect it to the **False** branch of the **Decision** activity. All transaction items that pass through this activity do not meet the condition added at step 7.
7. In the **Properties** panel, from the **Status** drop-down list, select **Successful**. This status is applied to all queue items that pass through this activity.
8. In the **TransactionItem** field, enter the QueueItem variable used at step 5.
9. (**Optionally**) Click the **Browse** browse button next to the **Output** field and enter the data that you want to be displayed in the **Item Details** window, in the **Output** field.

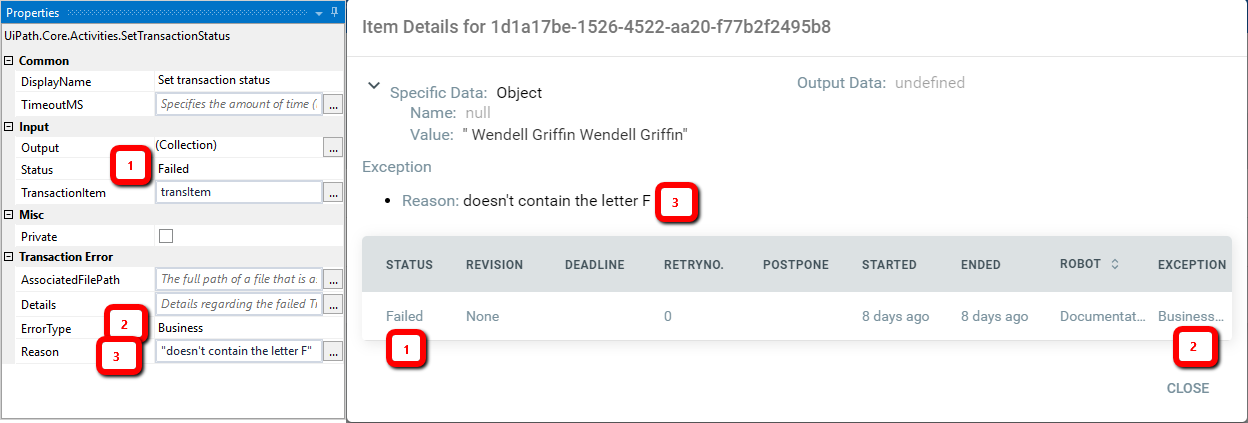
### Note:

This information is only displayed if the queue item is processed successfully.

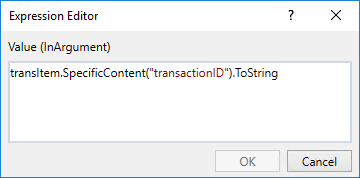
1. Drag another **Set Transaction Status** activity to the **Designer** panel and connect it to the **True** branch of the **Decision** activity. All queue items that pass through this activity meet the condition added at step 7.
2. In the **Properties** panel, from the **Status** field, select **Failed**. This status is applied to all queue items that pass through this activity.
3. In the **TransactionItem** field, enter the QueueItem variable used at step 5.
4. From the **ErrorType** list, select **Business** or **Application**, depending on the condition you put in the **Decision** activity. For example, an empty queue item could be an application exception, while a transaction value being smaller than a standard value could be a business exception. For more information on this choice, see the [Business Exception vs Application Exception](https://orchestrator.uipath.com/docs/business-exception-vs-application-exception) article.
5. In the **Reason** field, type a string to explain why the exception was triggered. After the project is executed, this information is displayed in the **Item Details** window.
6. Connect both **Set Transaction Status** activities to the **Get Transaction Item** activity. This creates a loop and enables you to iterate through all queue items. A basic automation project should look as in the following screenshot.

[](https://files.readme.io/a06bdd6-getting_queue_item_setting_status.png)

A mapping of the properties in the **Set Transaction Status** activity (on the left) and their corresponding fields in the **Item Details** window in Orchestrator.

[](https://files.readme.io/5fef190-mapping_status.png)

To retrieve specific data from a queue item, you can use the SpecificContent() function. Write the name of the argument that was used to add information to the queue between quotation marks in brackets.

[](https://files.readme.io/0ca6a0d-specific_content_from_queue.png)

For example, the screenshot above displays the **Value** field of an **Assign** activity. It enables you to retrieve the value attributed to a queue item (displayed in the **Item Details** window, under **Specific Data**, in Orchestrator) and assign it to a string variable, so that you can process it later.

In the screenshot above:

* transItem - is a QueueItem variable used to store transaction items.
* .SpecificContent - is the function that retrieves information from transaction items.
* ("transactionID") - is the name of the argument that was added in the **ItemInformation**window to populate the queue items with data. The argument name is also displayed in the **Item Details** window, under **Specific Data**, in Orchestrator.
* .ToString - is a function used to convert the ID of the queue, which was initially an integer, to a string.

**Videos:**

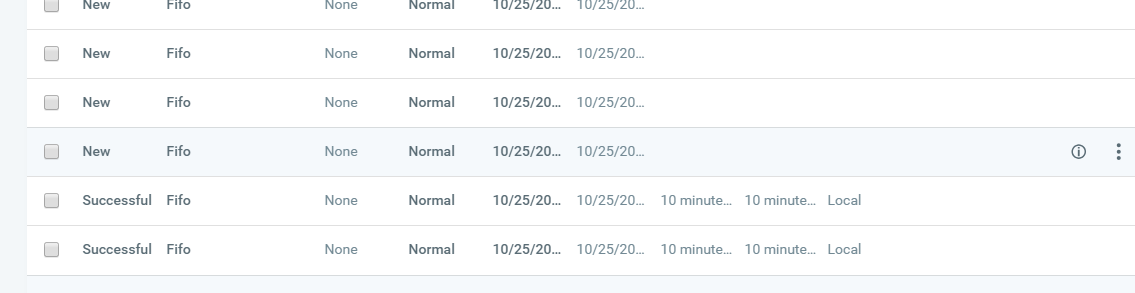
# UiPath Orchestrator | UiPath Tutorials | RPA Tutorial For Beginners | RPA Training | Edureka

[**https://www.youtube.com/watch?v=yY6GUoUNOD8&t=1808s**](https://www.youtube.com/watch?v=yY6GUoUNOD8&t=1808s)

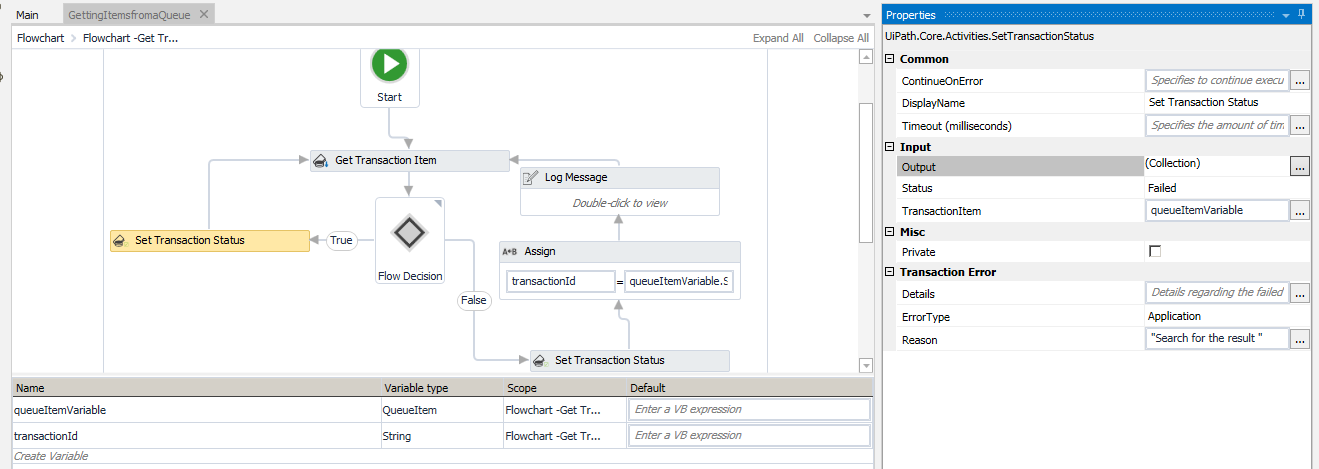
**Steps to Change the status of the item displayed in Orchestrator Queue**

**Precondition: create the Queue and make sure some items will be assigned with new status for the given Queue (Example: I have created Queue by Name “fifo” and on running schedular we have list of items with the status ‘New’ as showed in below screen shot**

**Schedular should be running to get the list of items in Queue and then run script from ui studio**

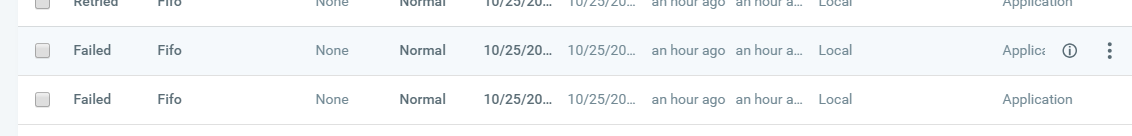


1. **Derived flow chart ( with 2 status ( Successful and Failure)**

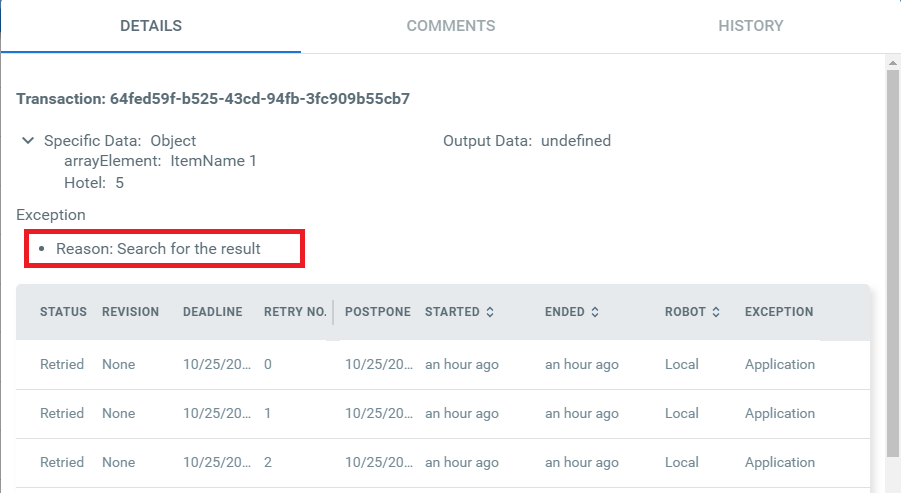


1. **When transaction flows towards True state , the specified status in properties is showed ( with appropriate reason message screen shot below)**

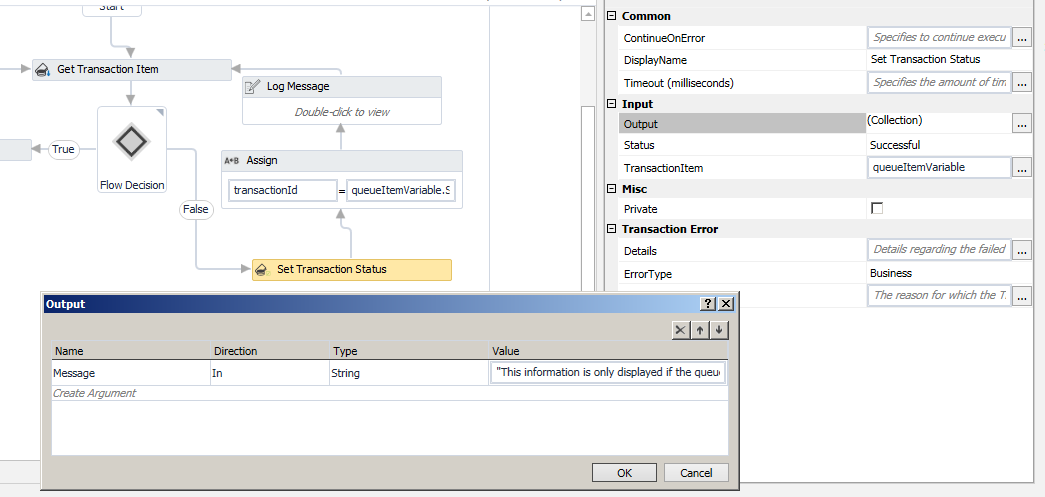
**On Failed Status change**



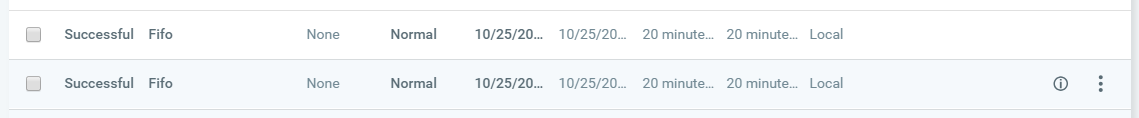
**Reason Mentioned / specified displayed**

****

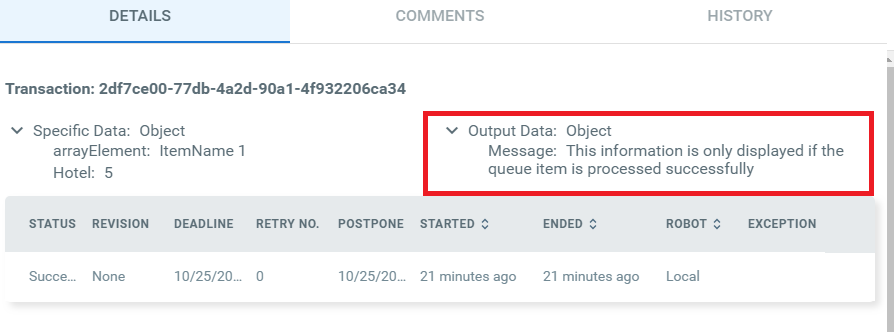
1. **On Passing transaction into False status specified message in the input is shown on Orchestrator**



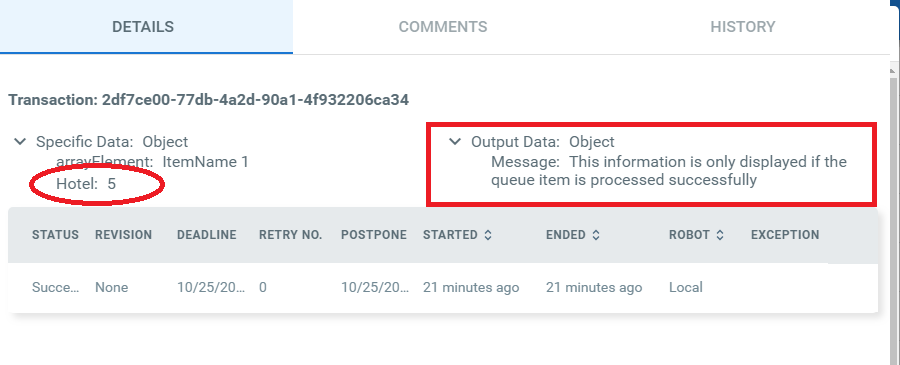
**On Successful status shown on item in orchestrator Queue**

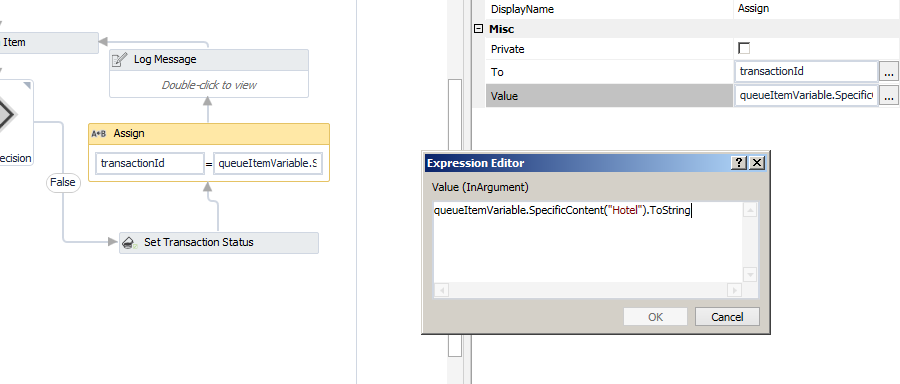


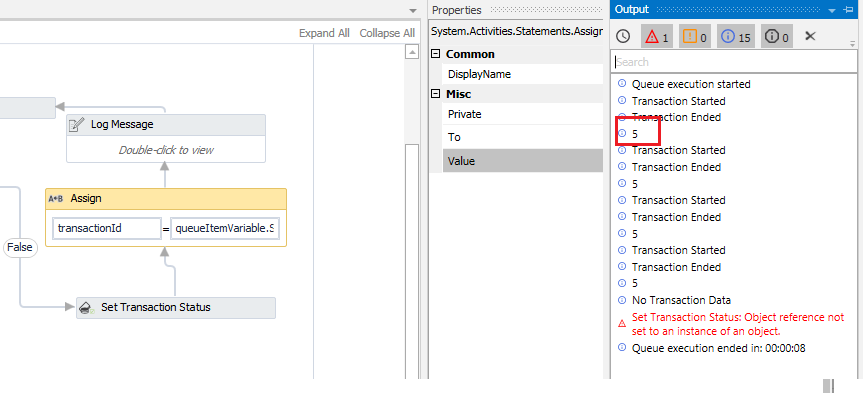
**Message displayed on transaction item on successful status**

****

**Getting Value of the specific item from Orchestrator to Ui studio ( Example : Hotel : 5, here key is Hotel and value is 5)**

****

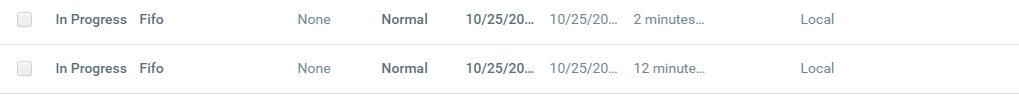


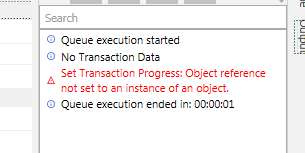
****

**Below shows the sample of xaml file ( which shows how to change the ‘status’ of item in orchestrator and getting the item value )**

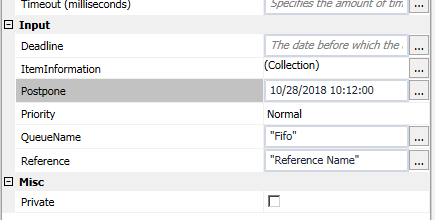
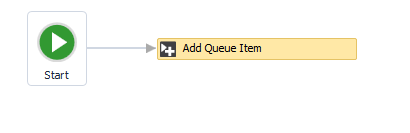
****

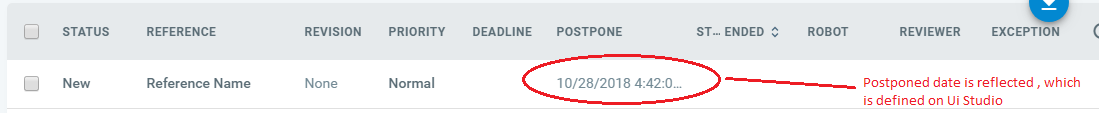
**If it is ‘in progress’ in orchestrator then there is ‘No Transaction Data’ on running Ui Studio on Get Transaction item**





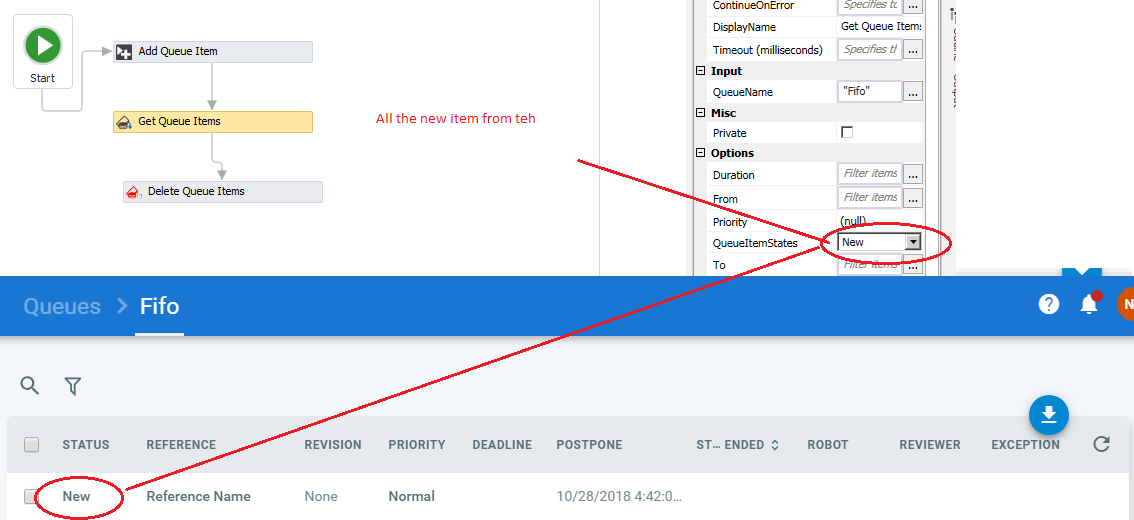
1. **Add Queue item**



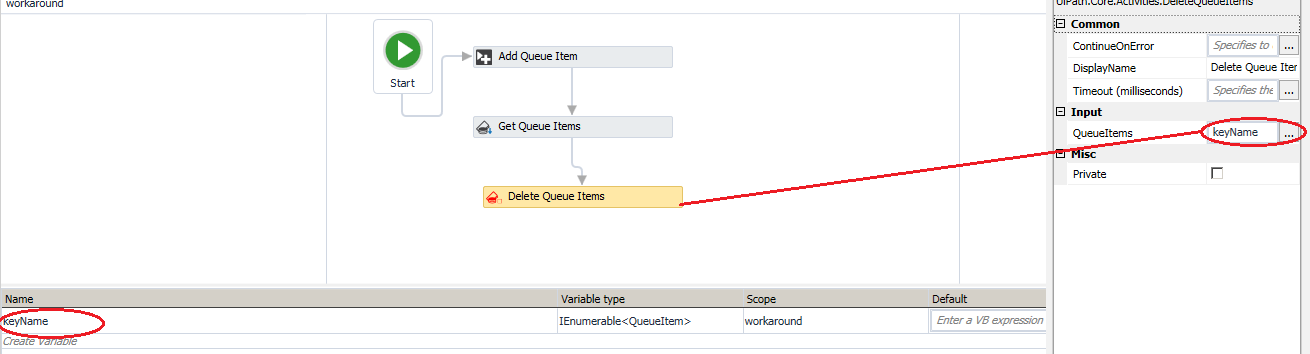
****

1. **Delete Queue Item**

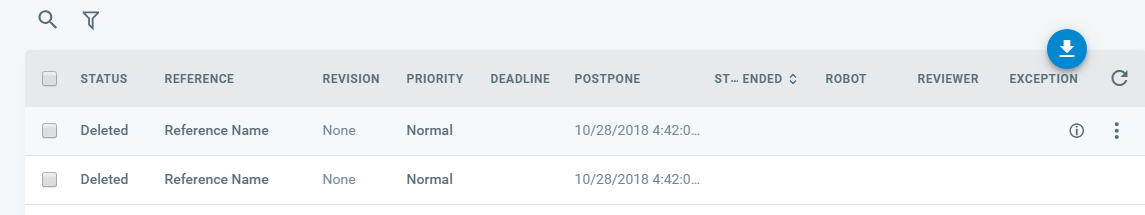
**All the item displayed in Orchestrator queue is deleted, based on the status provided**

****

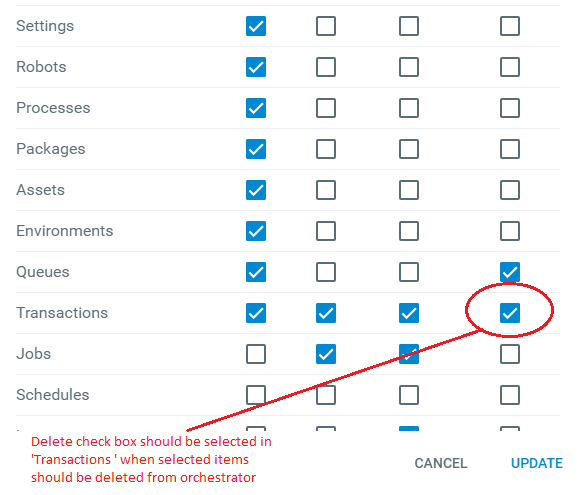
**Create the variable (example : keyName is below screen shot)**

****

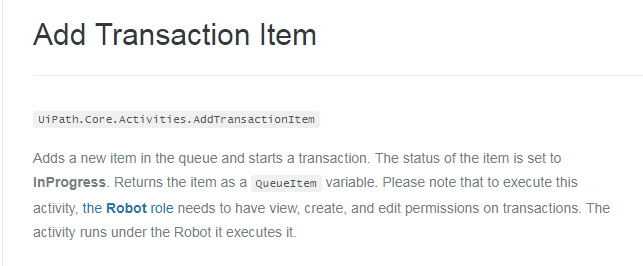
**After running the scrip the New status is changed to Delete Status**



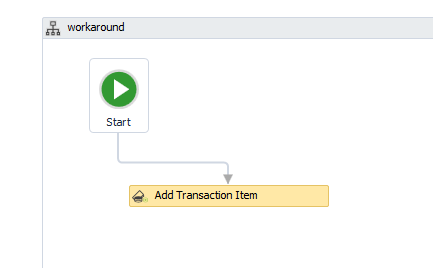
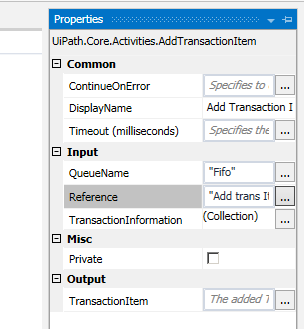
**Note : IF WE ARE GETTNIG ‘FORBIDDEN’ ERROR CHECH THE ‘DELETE’ PERMISSION FOR THE ‘TRANSACTION ‘ ROLE**

****

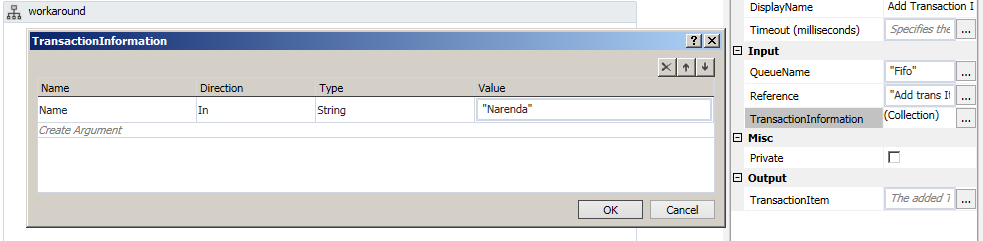
1. **Add Transaction Item**



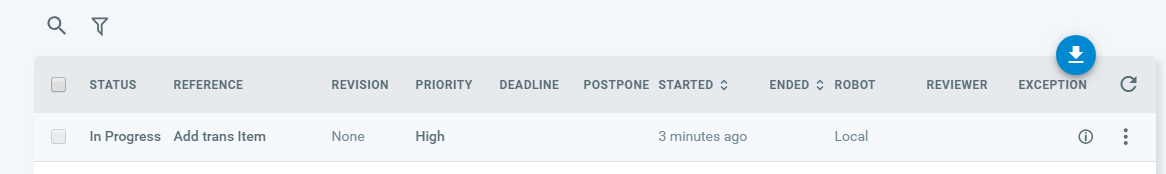
**Add Transaction Item Activity -> provide Queue Name (‘Fifo’ – specified in Orchestrator )**

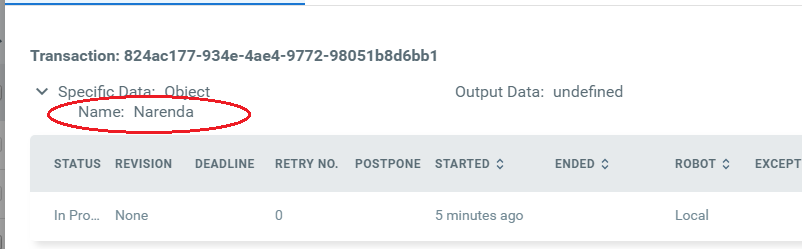
Add the Transaction Information



‘In Progress’ status is dispalyed on new created item



Defined Transaction infromation id displayed in item list



Difference between Add queue item and Add Transaction item

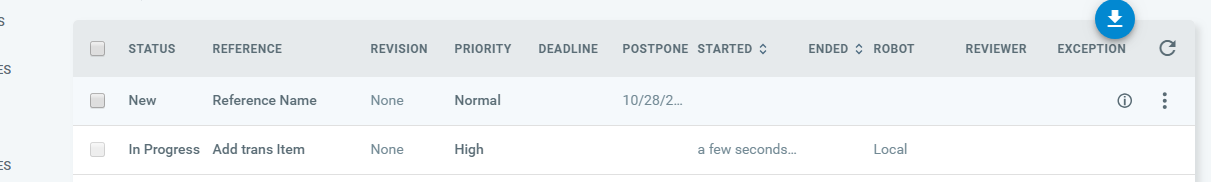
Queuen item will create the item with status **‘New** ‘ for the list of transactions in queue

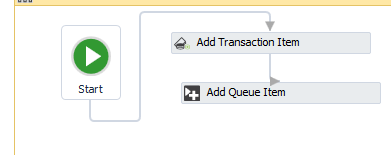
Transaction Item will create the item with status as ‘**In progress** ‘ with specific to single transaction

* **Add Queue Item** - populates the queue with items, defines a time frame when these items must be processed, and adds a custom reference to each item. By default, when an item is added to the queue its status is **New**.
* **Add Transaction Item** - adds an item in the queue, starts the transaction and sets the status to **In Progress**. It can also be configured to add a custom reference to each transaction.

Get Queue item - retrieve a list of transactions from an indicated queue  
So it has list of all transaction for the specified queue.

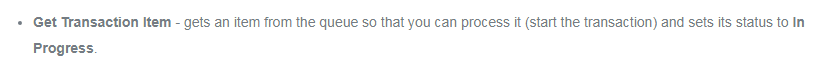
Get transaction - Gets an item from the queue  
It has only one transaction item.

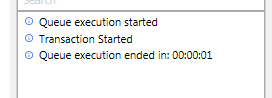
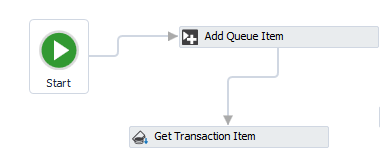




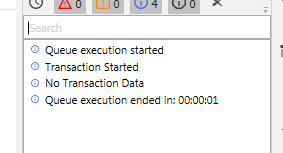
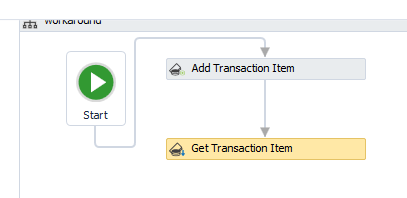
1. Get Transacation Item : will start with Status New and then convert to Inprogress

Steps : ( add Queue Item -> set Transaction -> run)



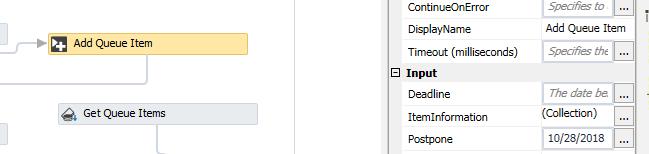


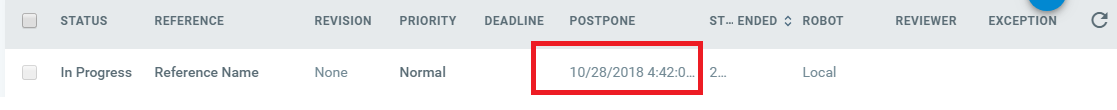
Note: If we add Add Transaction Item activity ( it create item with IN-Progress ) then adding Get Transaction item will result in No Transaction Data



1. **Postponed transaction (in the property the date is set to 10/28/2018) and the same is reflected in orchestrator**

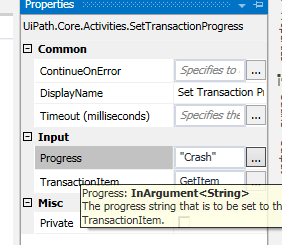
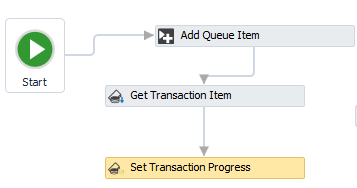
**Postpone Transaction Item** - adds time parameters between which a transaction must be processed (not before **Postpone** and no later than **Deadline**).

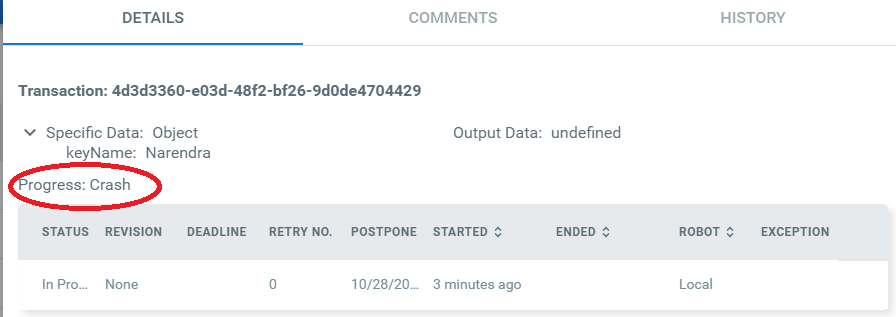


****

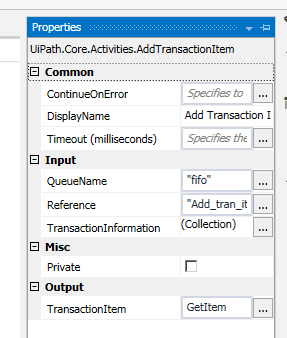
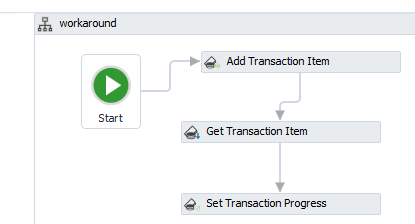
1. **Set Transaction Process**

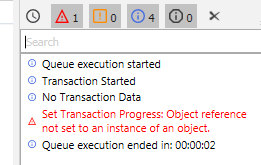
**Set Transaction Progress** - helps you create custom progress statuses for **In Progress** transactions. This can be placed in key places in your automation project, to inform you of its progress if, for example, it crashes. This option can be useful in the troubleshooting process.



****

**Note: On using Add Transaction Item: Error message thrown ‘Set Transaction Progress: Object reference not set to an instance of an object.’ (please find the screen shot below)**

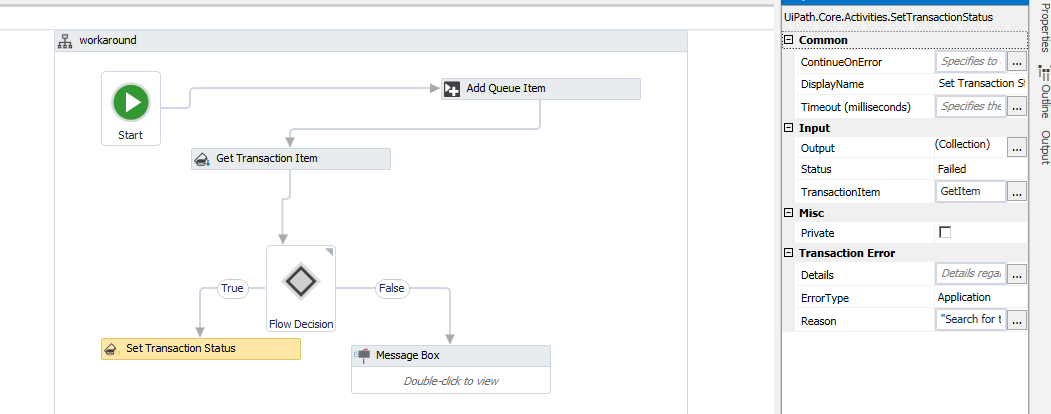


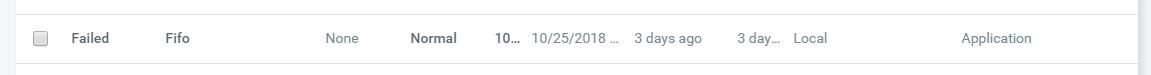


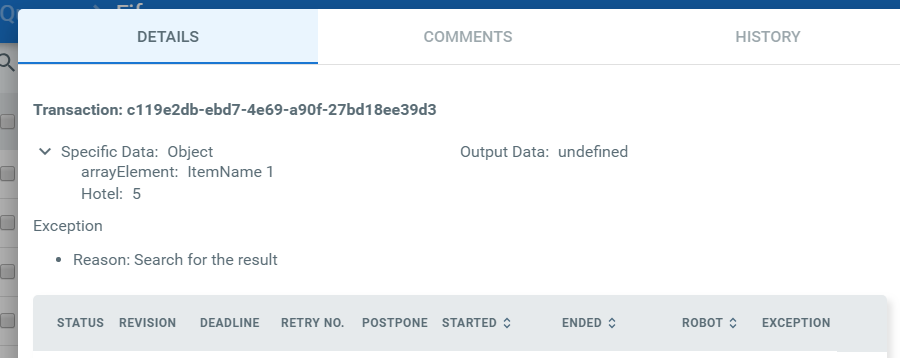
1. **Set Transaction Status**

**Set Transaction Status** - changes the status of the transaction item to **Failed** (with an **Application** or **Business** Exception) or **Successful**. Application Exceptions should be used when an app you are automating fails or is not responding, while Business Exceptions should be used when a certain transaction value does not meet a business requirement.

**Change the status of the queue from studio (see below screen shot for failed and reason specified)**







**Reference :**

**https://orchestrator.uipath.com/docs/studio-activities-used-with-queues**