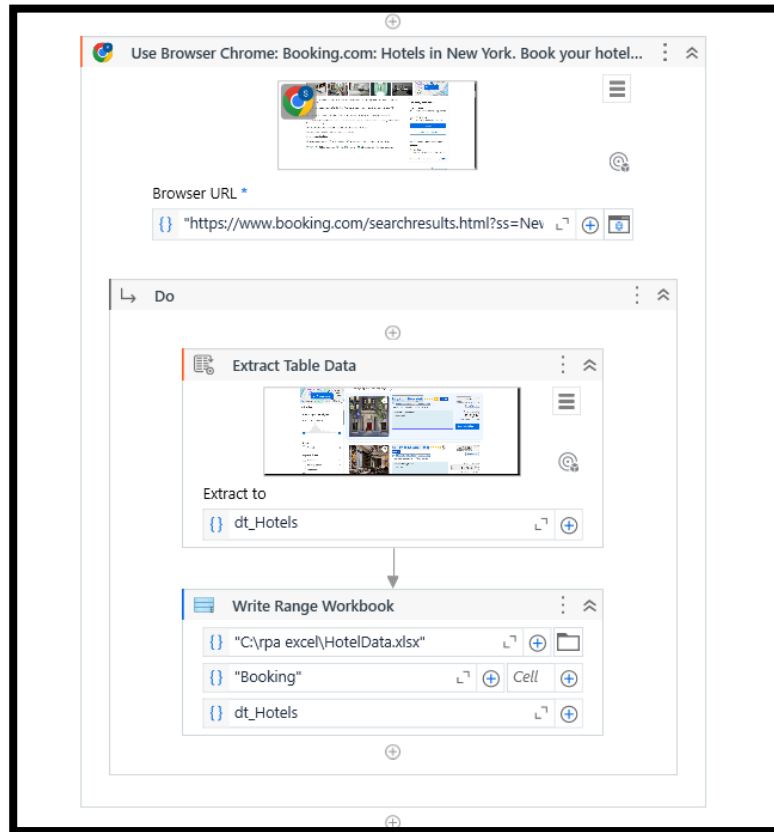


ROBOTIC PROCESS AUTOMATION (RPA)

Topic: Hotel Comparison Bot

Scrape_Booking.xaml:



The workflow demonstrates a simple web automation process that involves extracting data from the **Booking** website and writing it to an Excel file.

1. Use Browser Chrome

This activity is used to launch a Chrome browser and navigate to a specified URL. In this case, the URL is:

<https://www.booking.com/searchresults.html?ss=New+York>

This link opens the Booking.com search results page for hotels in New York. The browser session initiated here acts as the scope within which all subsequent web activities are performed.

2. Extract Table Data

Inside the browser scope, the "Extract Table Data" activity is used to scrape structured data from the web page. This typically includes tabular data such as hotel names, prices, ratings from the search results page.

The extracted data is stored in a DataTable variable named dt_Hotels. This variable temporarily holds the information retrieved from the web page for use in the next step.

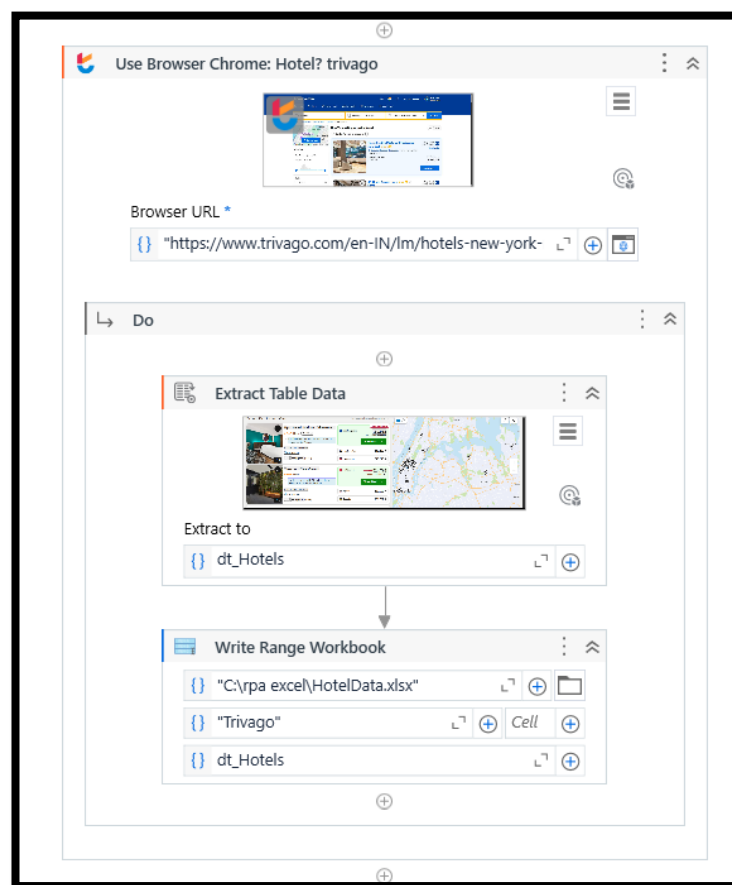
3. Write Range Workbook

The final step involves writing the extracted data to an Excel workbook. This is achieved using the "Write Range Workbook" activity. The key parameters used in this activity are as follows:

- **File Path:** C:\rpa excel\HotelData.xlsx – This is the location on the local system where the Excel file will be saved.
- **Sheet Name:** "Booking" – The name of the worksheet in which the data will be written.
- **DataTable:** dt_Hotels – The variable containing the data to be written.

This activity creates a new Excel file (or overwrites an existing one) and populates the specified worksheet with the hotel data that was extracted from the Booking.com web page.

Scrape_Trivago.xaml:



The workflow demonstrates a simple web automation process that involves extracting data from the **Trivago** website and writing it to an Excel file.

1. Use Browser Chrome

This activity is used to launch a Chrome browser and navigate to a specified URL. In this case, the URL is:

<https://www.trivago.com/en-IN/lm/hotels-new-york>

This link opens the Trivago search results page for hotels in New York. The browser session initiated here acts as the scope within which all subsequent web activities are performed.

2. Extract Table Data

Inside the browser scope, the "Extract Table Data" activity is used to scrape structured data from the web page. This typically includes tabular data such as hotel names, prices, ratings from the search results page.

The extracted data is stored in a DataTable variable named **dt_Hotels**. This variable temporarily holds the information retrieved from the web page for use in the next step.

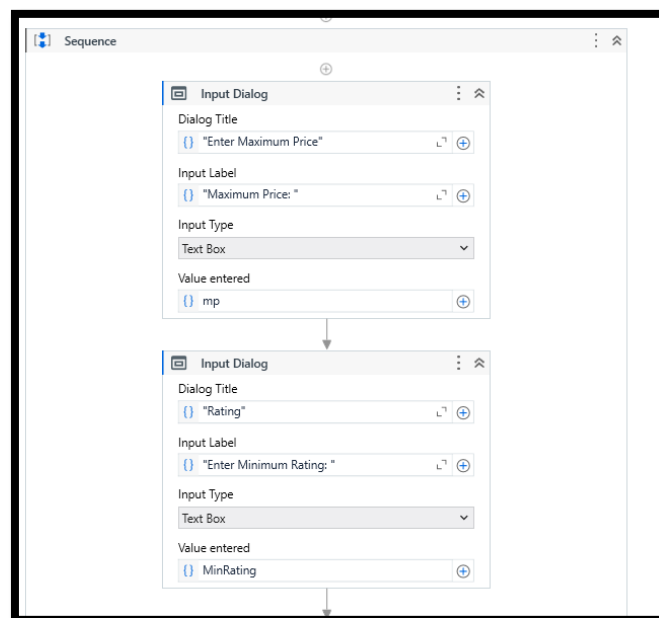
3. Write Range Workbook

The final step involves writing the extracted data to an Excel workbook. This is achieved using the "Write Range Workbook" activity. The key parameters used in this activity are as follows:

- **File Path:** C:\rpa excel\HotelData.xlsx – This is the location on the local system where the Excel file will be saved.
- **Sheet Name:** "Trivago" – The name of the worksheet in which the data will be written.
- **DataTable:** dt_Hotels – The variable containing the data to be written.

This activity creates a new Excel file (or overwrites an existing one) and populates the specified worksheet with the hotel data that was extracted from the Trivago web page.

Main.xaml:



This demonstrates how to incorporate user inputs for dynamic data filtering within an RPA (Robotic Process Automation) process. It involves two **Input Dialog** activities used to collect user preferences regarding hotel listings. These inputs are later used to filter the extracted data based on price and rating.

1. Input Dialog: Enter Maximum Price

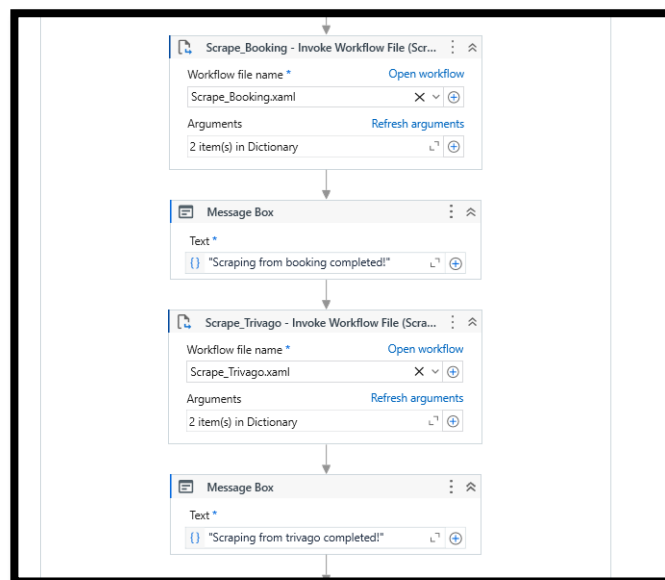
The first step in the sequence presents the user with an input dialog titled **"Enter Maximum Price."** This dialog prompts the user to input a value that represents the highest price they are willing to pay for a hotel room. The prompt is displayed through the label **"Maximum Price: "** and uses a **Text Box** as the input type, allowing the user to enter numerical data freely.

The entered value is stored in a variable named **mp**, which will later be used in the automation process to filter out hotels that exceed the user's specified budget.

2. Input Dialog: Enter Minimum Rating

Following the price input, the next dialog box is titled **"Rating"**, with the prompt **"Enter Minimum Rating: "** displayed to the user. This dialog similarly uses a **Text Box** input field, allowing the user to specify the minimum acceptable rating for a hotel. The value entered is captured in a variable named **MinRating**. This input will assist in filtering the dataset so that only hotels meeting or exceeding the user's rating expectations are retained.

The primary purpose of these input dialogs is to allow dynamic customization of the hotel data extraction process based on individual user preferences. Once the data is extracted from hotel booking websites such as Booking.com or Trivago, these user-defined variables (**mp** for maximum price and **MinRating** for minimum acceptable rating) are used as **filters** in subsequent steps. This ensures that the final dataset only includes hotels that meet the user's budget and quality expectations. Such interactivity enhances the flexibility and user-centric nature of the automation workflow, allowing the same process to be reused across various scenarios with different user requirements. It also supports better decision-making by providing filtered, relevant data rather than raw, unstructured information.



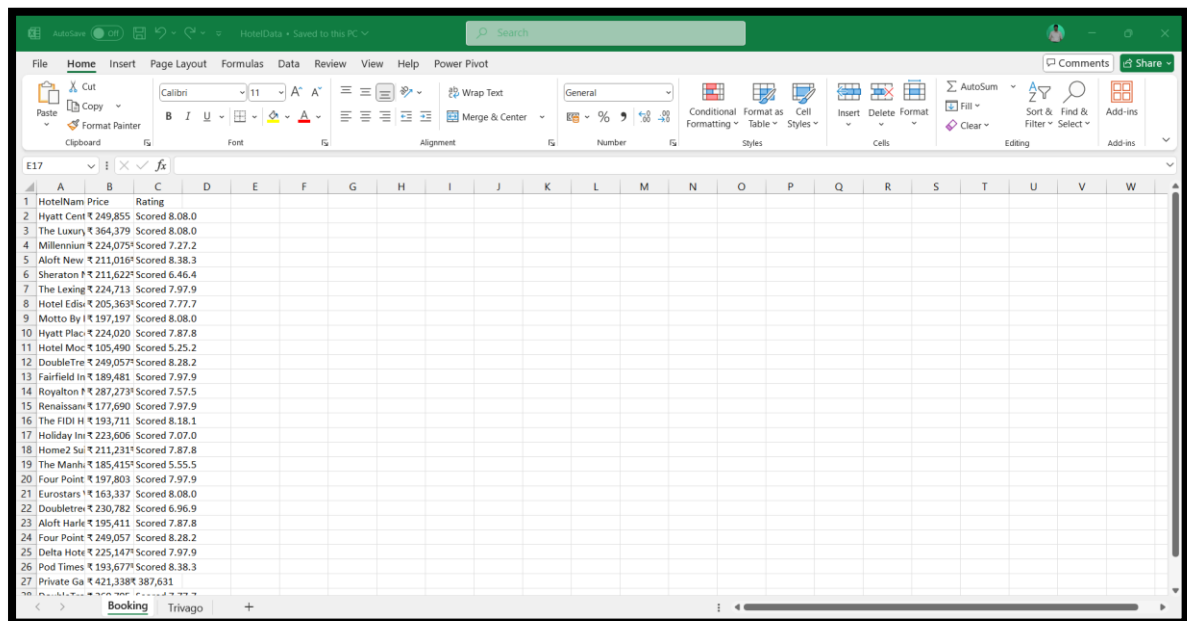
This manages the centralized execution of two separate scraping workflows—**Booking.com** and **Trivago**—ensuring efficient and organized data extraction.

1. Invoke Scrape_Booking.xml

Executes the Booking.com scraping workflow using the provided input parameters (maximum price, minimum rating). Data is extracted and saved to an Excel file.

2. Message Box – “Scraping from booking completed!”

Confirms successful completion of the Booking.com data extraction.



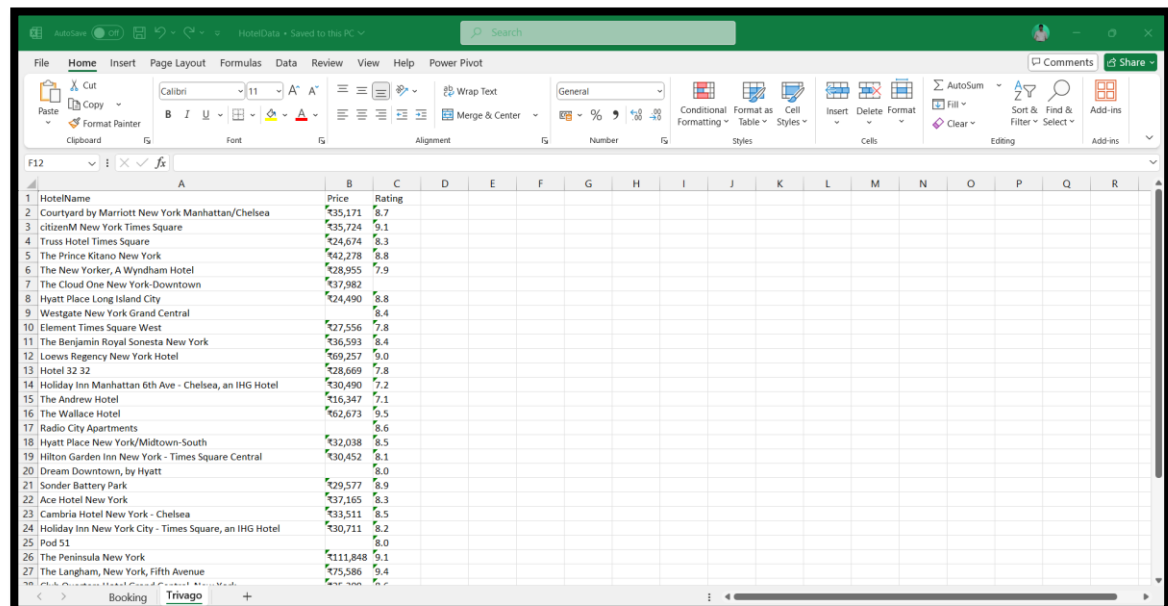
| HotelName | Price | Rating |
|--------------|-----------|----------------|
| Hyatt Cent | ₹ 249,855 | Scored 8.08.0 |
| The Luxury | ₹ 364,379 | Scored 8.08.0 |
| Millennium | ₹ 224,075 | Scored 7.27.2 |
| Aloft New | ₹ 211,016 | Scored 8.38.3 |
| Sheraton | ₹ 211,622 | Scored 6.46.4 |
| The Lexing | ₹ 224,713 | Scored 7.97.9 |
| Hotel Edis | ₹ 205,363 | Scored 7.77.7 |
| Motto By I | ₹ 197,197 | Scored 8.08.0 |
| Hyatt Plac | ₹ 224,020 | Scored 7.87.8 |
| Hotel Moc | ₹ 105,490 | Scored 5.25.2 |
| DoubleTre | ₹ 249,057 | Scored 8.28.2 |
| Fairfield In | ₹ 189,481 | Scored 7.97.9 |
| Royalton | ₹ 287,273 | Scored 7.57.5 |
| Renaissanc | ₹ 177,690 | Scored 7.97.9 |
| The FIDI H | ₹ 193,711 | Scored 8.18.1 |
| Holiday Im | ₹ 223,606 | Scored 7.07.0 |
| Home2 Su | ₹ 211,231 | Scored 7.87.8 |
| The Manh | ₹ 185,415 | Scored 5.55.5 |
| Four Point | ₹ 197,803 | Scored 7.97.9 |
| Eurostars | ₹ 163,337 | Scored 8.08.0 |
| Doubletre | ₹ 230,782 | Scored 6.96.9 |
| Aloft Harle | ₹ 195,411 | Scored 7.87.8 |
| Four Point | ₹ 249,057 | Scored 8.28.2 |
| Delta Hot | ₹ 225,147 | Scored 7.97.9 |
| Pod Times | ₹ 193,677 | Scored 8.38.3 |
| Private Ga | ₹ 421,338 | Scored 387.631 |

3. Invoke Scrape_Trivago.xaml

Executes the Trivago scraping workflow similarly, using the same input structure.

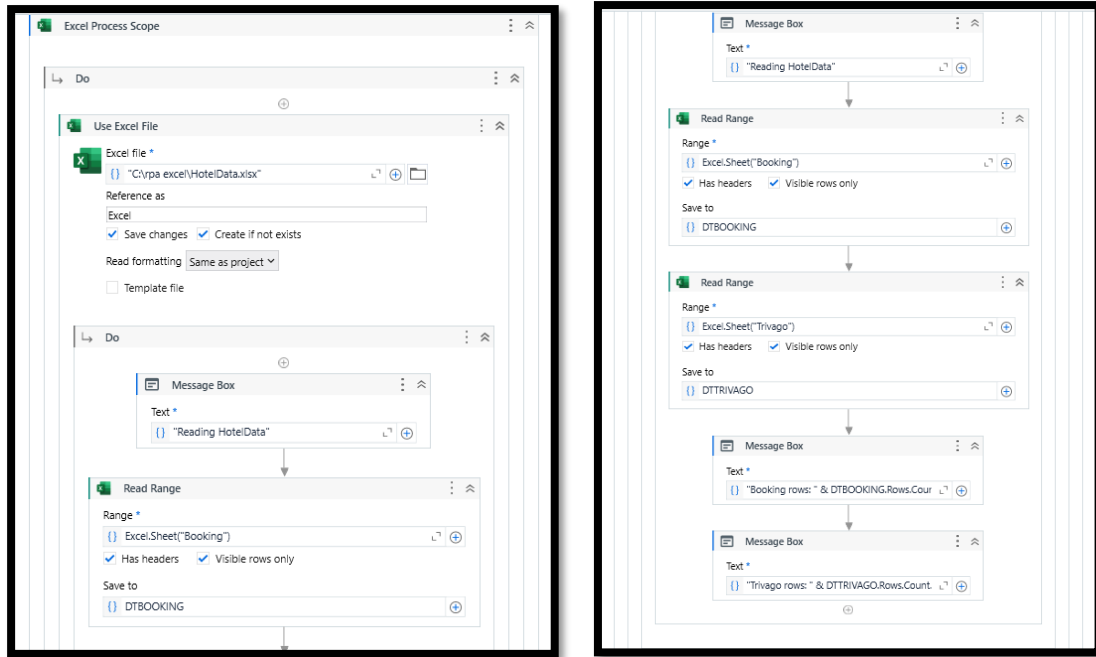
4. Message Box – “Scraping from trivago completed!”

Confirms successful Trivago data extraction.



| HotelName | Price | Rating |
|--|---------|--------|
| Courtyard by Marriott New York Manhattan/Chelsea | ₹35,171 | 8.7 |
| citizenM New York Times Square | ₹35,724 | 9.1 |
| Truss Hotel Times Square | ₹24,674 | 8.3 |
| The Prince Kitano New York | ₹42,278 | 8.8 |
| The New Yorker, A Wyndham Hotel | ₹28,955 | 7.9 |
| The Cloud One New York-Downtown | ₹37,982 | 8.6 |
| Hyatt Place Long Island City | ₹24,490 | 8.4 |
| Westgate New York Grand Central | ₹27,556 | 7.8 |
| Element Times Square West | ₹36,593 | 8.4 |
| The Benjamin Royal Sonesta New York | ₹69,257 | 9.0 |
| Loews Regency New York Hotel | ₹28,669 | 7.8 |
| Hotel 32 32 | ₹30,490 | 7.2 |
| Holiday Inn Manhattan 6th Ave - Chelsea, an IHG Hotel | ₹16,347 | 7.1 |
| The Andrew Hotel | ₹62,673 | 9.5 |
| The Wallace Hotel | ₹32,038 | 8.5 |
| Radio City Apartments | ₹30,452 | 8.1 |
| Hyatt Place New York/Midtown-South | ₹29,577 | 8.9 |
| Hilton Garden Inn New York - Times Square Central | ₹37,165 | 8.3 |
| Dream Downtown, by Hyatt | ₹33,511 | 8.5 |
| Sonder Battery Park | ₹30,711 | 8.2 |
| Ace Hotel New York | ₹11,848 | 9.1 |
| Cambria Hotel New York - Chelsea | ₹75,586 | 9.4 |
| Holiday Inn New York City - Times Square, an IHG Hotel | ₹30,711 | 8.2 |
| Pod 51 | ₹11,848 | 9.1 |
| The Peninsula New York | ₹75,586 | 9.4 |
| The Langham, New York, Fifth Avenue | ₹30,711 | 8.2 |

This demonstrates modular and reusable workflow design. It automates hotel data collection from multiple sources, providing structured data for analysis and comparison. The approach supports scalability, allowing easy addition of new sources in the future.



This is responsible for reading the hotel data previously scraped from Booking.com and Trivago, and displaying the number of entries extracted from each source. It uses UiPath's **Excel Process Scope** and activities for structured file operations.

1. **Excel Process Scope:**

Opens the Excel file located at C:\rpa excel\HotelData.xlsx, allowing subsequent activities to interact with the file.

2. **Read Booking Data:**

- A **Message Box** first displays “Reading HotelData” to indicate the beginning of the read operation.
- A **Read Range** activity reads all visible rows from the "Booking" sheet (with headers).
- The data is stored in a DataTable named DTBOOKING.

3. **Read Trivago Data:**

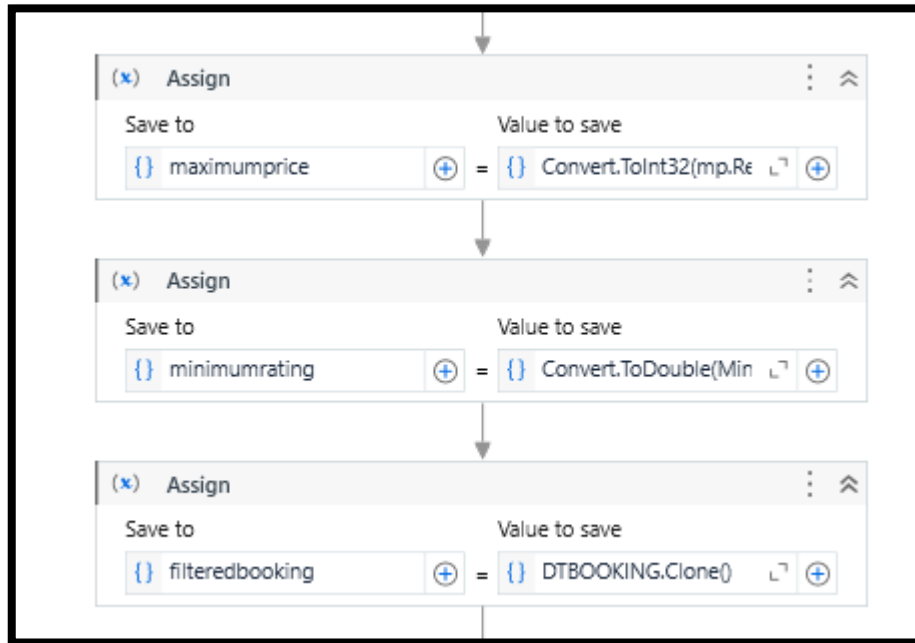
- Another **Read Range** activity reads visible rows from the "Trivago" sheet.
- This data is saved to a DataTable named DTTRIVAGO.

4. **Count and Display Records:**

- Two **Message Box** activities are used to show the number of rows retrieved from each sheet:

- One for Booking: "Booking rows: " & DTBOOKING.Rows.Count.ToString
- One for Trivago: "Trivago rows: " & DTTRIVAGO.Rows.Count.ToString

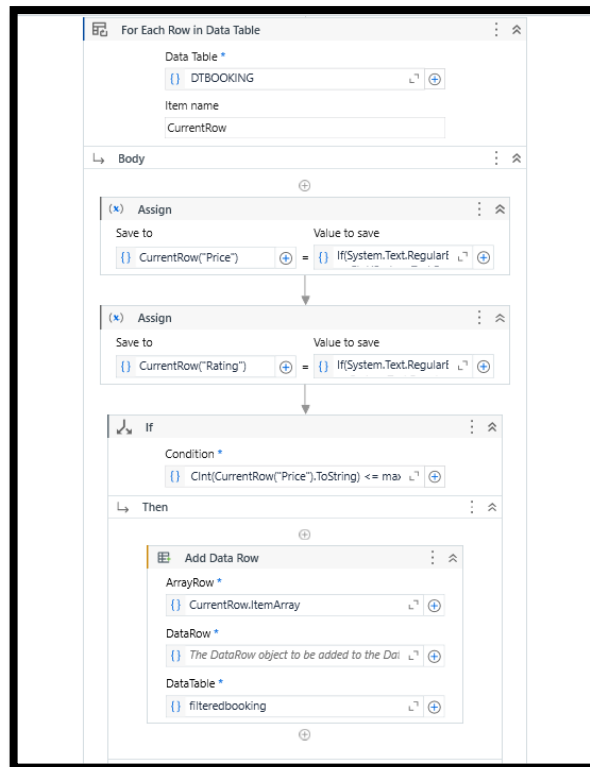
This workflow ensures that data from both Booking.com and Trivago has been successfully written to and can be retrieved from the Excel file. It validates the scraping results by displaying the row counts for both platforms.



This segment prepares the data for filtering based on user input.

1. **Extract Maximum Price:**
The variable maximumprice is set by converting the user input (mp) into an integer, removing any commas and extra spaces.
2. **Extract Minimum Rating:**
The variable minimumrating is set by converting the user input (MinRating) into a double, ensuring correct numerical format.
3. **Initialize Filtered Data Table:**
A new DataTable filteredbooking is created as a clone of DTBOOKING, replicating its structure (columns) but with no data. This will later be used to store only the filtered results that meet the specified price and rating conditions.

This setup ensures that only hotels from Booking.com that meet the user's price and rating criteria will be selected and stored for further comparison.



A **For Each Row** activity is used to iterate through each row of the DTBOOKING DataTable. The goal is to extract and standardize the "Price" and "Rating" values, and filter out hotels that do not meet the user's specified criteria.

1. Price Cleaning and Conversion:

The "Price" value may contain comma-separated currency formats (e.g., ₹1,200). Using a regular expression, the last matched numeric pattern is extracted and the commas are removed. The resulting value is converted to an integer. If no valid numeric pattern is found, the price is defaulted to 0.

2. Rating Extraction:

The "Rating" field is parsed to extract the first decimal number format (e.g., 4.3). If no valid rating is found, the value is defaulted to "0".

3. Filter Condition:

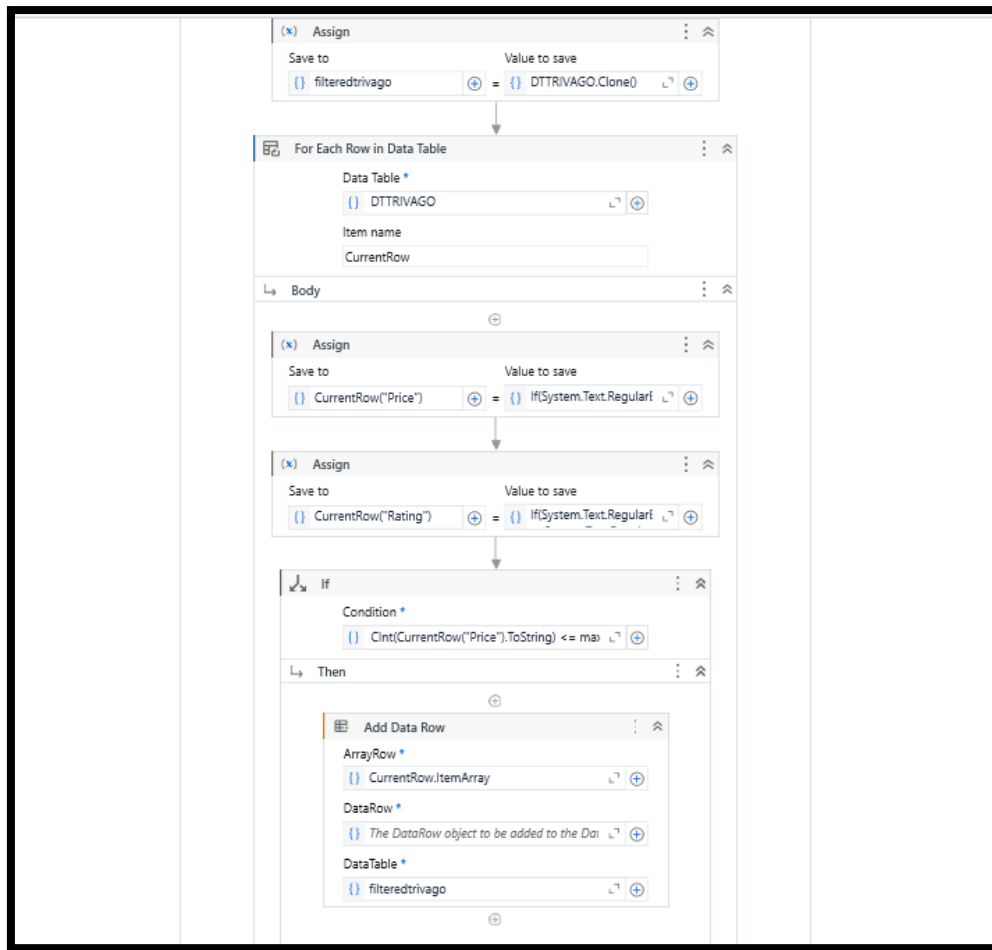
An If condition is used to filter hotels based on user-defined thresholds:

- The cleaned Price must be **less than or equal to** the specified maximumprice.
- The extracted Rating must be **greater than or equal to** the specified minimumrating.

4. Storing Filtered Data:

If a row satisfies both conditions, it is added to a new DataTable named filteredbooking using the Add Data Row activity.

This logic ensures only the hotels matching the price and rating preferences of the user are retained for further comparison or display.



This filters hotel listings from the DTTRIVAGO DataTable based on user-defined price and rating thresholds, and stores the matching results in a new DataTable called filteredtrivago.

1. Cloning the Structure:

A new DataTable filteredtrivago is created by cloning the structure of DTTRIVAGO. This ensures the filtered data maintains the same column format.

2. Iterating Through Rows:

The For Each Row activity loops through every row of the DTTRIVAGO DataTable, with the variable CurrentRow representing the current row in each iteration.

3. Price Formatting and Conversion:

Using a regular expression, the workflow extracts the last valid numeric price from the "Price" column (e.g., ₹2,500 → 2500). Any commas are removed, and the value is converted to an integer. If the price format is not matched, it defaults to 0.

4. Rating Extraction:

The "Rating" value is extracted using a regex that identifies the first decimal value (e.g., 4.3). If the format is not matched, it is set to "0" by default.

5. Applying Filter Criteria:

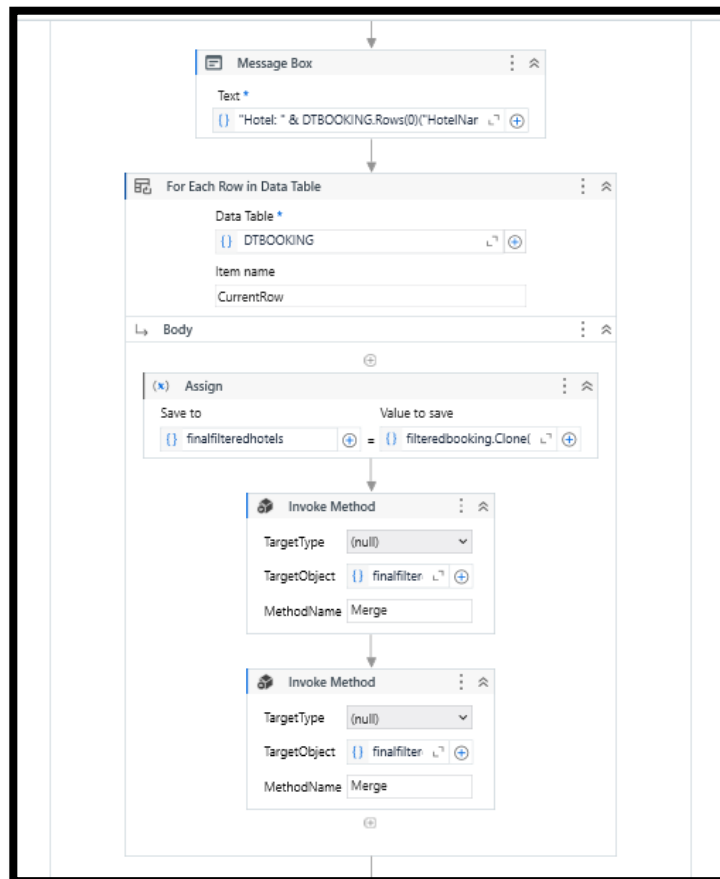
An If activity is used to check:

- Whether the extracted price is **less than or equal to** the user-defined maximumprice.
- Whether the extracted rating is **greater than or equal to** the user-defined minimumrating.

Only rows that satisfy both conditions proceed to the next step.

6. Adding Matched Rows:

Rows meeting the above criteria are added to the filteredtrivago DataTable using the Add Data Row activity, with CurrentRow.ItemArray used to copy the entire row.



Parameters ? □ ✕

—

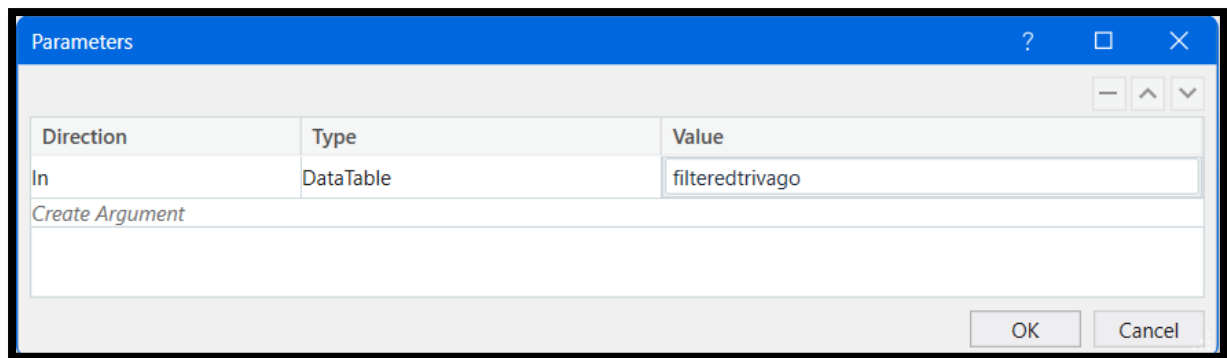
^

▼

| Direction | Type | Value |
|-----------|-----------|-----------------|
| In | DataTable | filteredbooking |

Create Argument

OK Cancel



After filtering hotel data from both Booking.com and Trivago based on user-defined criteria (maximum price and minimum rating), the next step involves merging the filtered results into a single unified DataTable. This ensures that the user receives a consolidated list of all qualifying hotel options across platforms.

Displaying Sample Data

To verify that data has been correctly extracted, a Message Box is used to display the details of the first hotel entry from the Booking.com DataTable (DTBOOKING). The message displays the hotel name, price, and rating in a formatted string to help validate data integrity before proceeding with merging operations.

Initializing the Final DataTable

A new DataTable called `finalfilteredhotels` is initialized using the `Clone()` method on the `filteredbooking` DataTable. The `Clone()` function copies only the structure (column headers) of the original table, allowing us to create an empty DataTable with the same format.

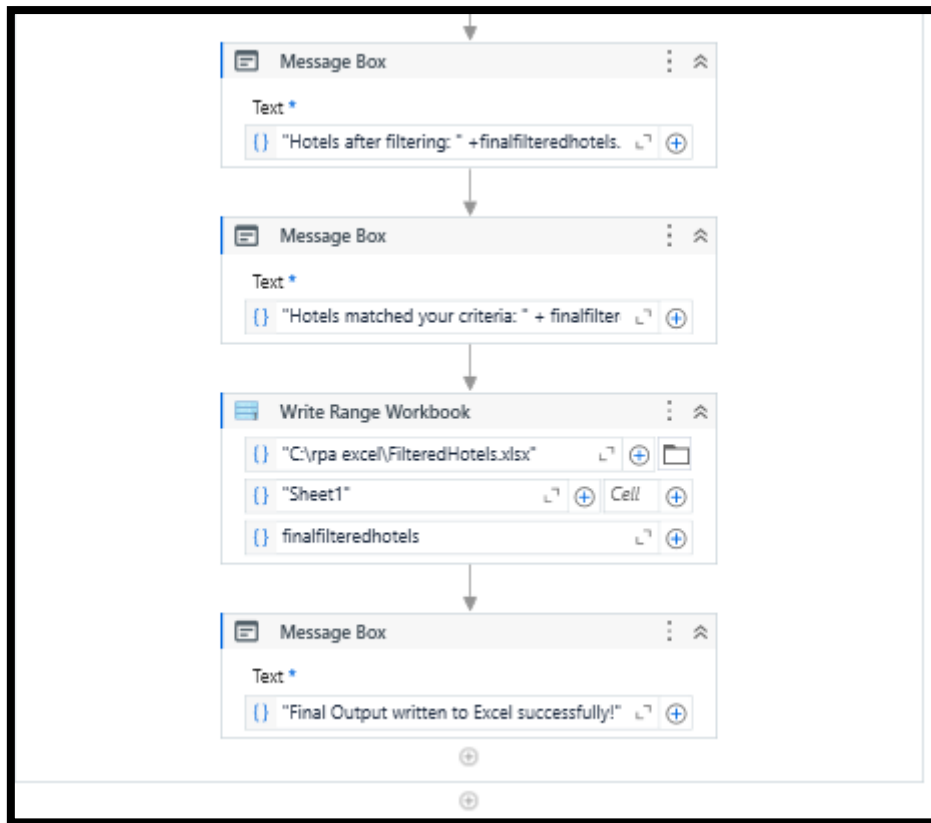
Merging Filtered Results

Two Invoke Method activities are used to merge data into a DataTable named `finalfilter`. This process combines the filtered hotel records from both:

- `filteredbooking` (filtered hotels from Booking.com)
- `filteredtrivago` (filtered hotels from Trivago)

The Merge method is called in each case to append the rows from each source into the `finalfilter` DataTable. Both Invoke Method activities are configured to receive a DataTable as input (Direction: **In**, Type: **DataTable**), and their values are set respectively to `filteredbooking` and `filteredtrivago`.

This merging step ensures that all qualifying hotels, regardless of source, are brought into a single table. The final result (`finalfilteredhotels`) can then be exported, visualized, or used for further processing. It allows for a seamless and platform-agnostic hotel comparison experience for the end user.



Once the filtered and merged hotel data is stored in the finalfilteredhotels DataTable, the process proceeds to inform the user of the results and export them for reference.

Message Box: Displaying Filtered Results Count

Two message boxes are used to display the number of hotels that meet the user's search criteria:

1. First Message Box

- **Text:** "Hotels after filtering: " + finalfilteredhotels.Rows.Count.ToString
- **Purpose:** This shows the total number of hotels present in the finalfilteredhotels DataTable after merging and filtering operations. It confirms that the merging logic worked and gives the user an idea of how many options are available.

2. Second Message Box

- **Text:** "Hotels matched your criteria: " + finalfilteredhotels.Rows.Count.ToString
- **Purpose:** This is essentially a reiteration of the previous message, but more user-friendly. It assures the user that these hotels matched their specified criteria (such as location, price, and rating).

Exporting to Excel

The Write Range Workbook activity is used to export the final filtered results to an Excel file for easy viewing and sharing:

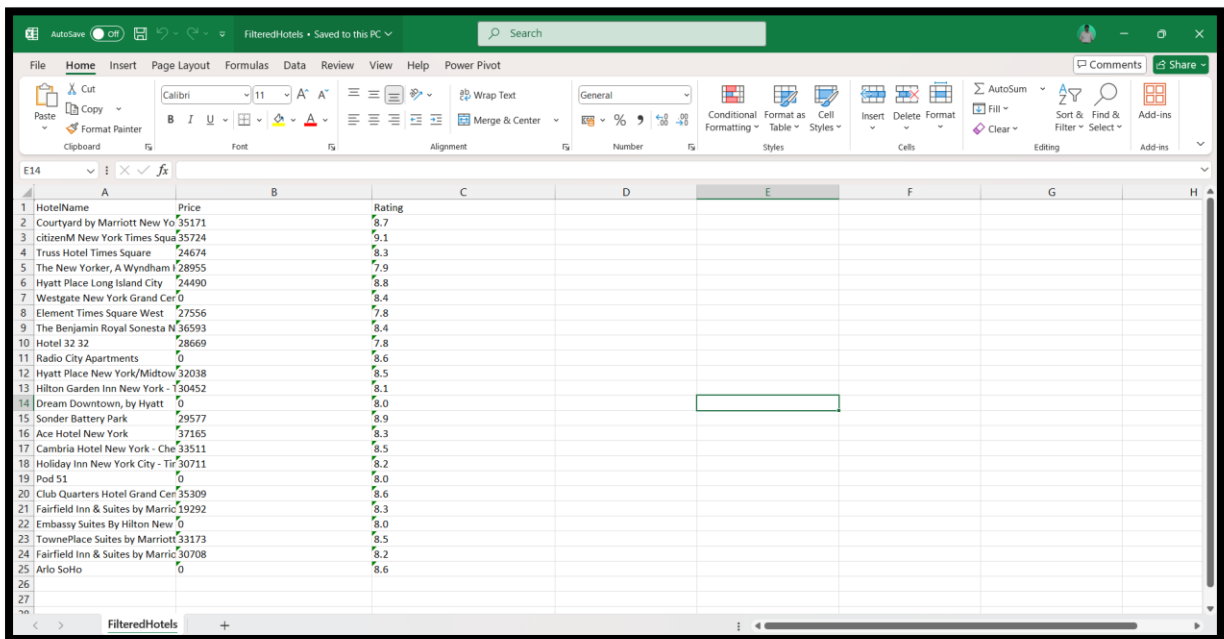
- **File Path:** C:\rpa excel\FilteredHotels.xlsx
- **Sheet Name:** "Sheet1"

- **Data Table:** finalfilteredhotels

This activity writes the entire contents of the finalfilteredhotels DataTable to an Excel spreadsheet, including all columns and rows.

Confirmation Message Box

- **Text:** "Final Output written to Excel successfully!"
- **Purpose:** Provides confirmation to the user that the data has been successfully written to the specified Excel file, completing the hotel comparison workflow.



The screenshot shows an Excel spreadsheet titled "FilteredHotels - Saved to this PC". The spreadsheet contains a table with three columns: "HotelName", "Price", and "Rating". The data is as follows:

| HotelName | Price | Rating |
|---------------------------------|--------|--------|
| Courtyard by Marriott New Yo | 35171 | 8.7 |
| citizenM New York Times Squa | 35724 | 9.1 |
| Truss Hotel Times Square | 24674 | 8.3 |
| The New Yorker, A Wyndham | 28955 | 7.9 |
| Hyatt Place Long Island City | 24490 | 8.8 |
| Westgate New York Grand Cen | 0 | 8.4 |
| Element Times Square West | 27556 | 7.8 |
| The Benjamin Royal Sonesta N | 36593 | 8.4 |
| Hotel 32 32 | 28669 | 7.8 |
| Radio City Apartments | 0 | 8.6 |
| Hyatt Place New York/Midtow | 32038 | 8.5 |
| Hilton Garden Inn New York - | 130452 | 8.1 |
| Dream Downtown, by Hyatt | 0 | 8.0 |
| Sonder Battery Park | 29577 | 8.9 |
| Ace Hotel New York | 37165 | 8.3 |
| Cambria Hotel New York - Che | 33511 | 8.5 |
| Holiday Inn New York City - Tir | 30711 | 8.2 |
| Pod 51 | 0 | 8.0 |
| Club Quarters Hotel Grand Cen | 35309 | 8.6 |
| Fairfield Inn & Suites by Marri | 19292 | 8.3 |
| Embassy Suites By Hilton New | 0 | 8.0 |
| TownePlace Suites by Marriott | 33173 | 8.5 |
| Fairfield Inn & Suites by Marri | 30708 | 8.2 |
| Arlo SoHo | 0 | 8.6 |