

Business Process Automation

Ü3 RPA Implementation with UiPath

WS 23/24

09.01.2024

Sehej Bhasin (605927)
Yannick Elß (636477)
Michal Gorecki (636698)
Vanessa Ernst (599755)

For the RPA implementation in our movie reservation process with UiPath, we select a fragment of the model from Ü1 and adjusted it to fulfill the model requirements. We decided to simplify the model by removing certain components that were not essential for the implementation of the file handling, screen scraping, browser actions and loop.

The implemented process starts with decision point, model with while loop. The condition of the loop is based on a predefined string variable “User_Ready” initially set to “no.” Within this loop, the system prompts the customer with a query about their readiness to proceed with the ticket booking. This loop continues until the customer indicates readiness by responding with “yes.” Upon receiving this affirmative response, the loop terminates, triggering the subsequent browser action. The corresponding Input Dialog used for the implementation is shown in Figure 1.

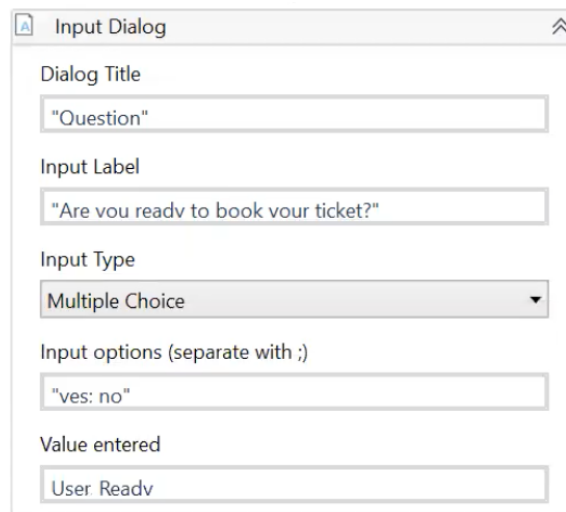
The image shows a screenshot of the 'Input Dialog' configuration window in UiPath. The window has a title bar with a standard icon and a maximize button. Inside, there are several labeled fields: 'Dialog Title' with the value 'Question', 'Input Label' with the value 'Are vou readv to book vour ticket?', 'Input Type' set to 'Multiple Choice' (indicated by a dropdown arrow), 'Input options (separate with :)' with the value 'ves: no', and 'Value entered' with the value 'User Readv'. The text in the input fields appears to be a mix of correct and incorrect characters, possibly due to OCR or a typo in the original image.

Figure 1: Input Dialog of While Loop in UiPath

We combined the screen scraping with the browser action, by scraping the reservation of a seat from <https://www.UCI-kinowelt.de>. Once the browser action is initiated, a "Do" activity is implemented. It includes the maximalization of the browser window to ensure that all necessary elements on the website are visible and accessible for subsequent actions. Following this, we have incorporated a delay of 15 seconds. This pause allows enough time for the browser to load the website completely to ensure the success of the screen scraping and browser interaction activities that follow. After the website is fully loaded, the automation script proceeds to scroll down the website to reach the section of the website where seat reservation options are available. Once the desired page section is in view, the script automatically identifies and selects the "Als Gast buchen" (Book as Guest) button. Following this, it enters a user email address using the 'Type Input' activity, as illustrated in Figure 2.

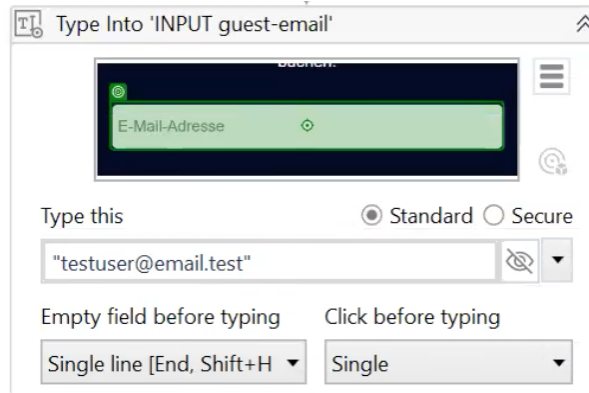


Figure 3: Automating Text Input in UiPath

After a brief delay of 3 seconds, the script clicks the “Weiter ohne mein UCI” (Continue without UCI account) button. This is followed by another short delay, after which it selects the ticket type designated for adults. The script continues to scroll down the website to access the seat selection area. After a further delay, a seat is chosen, and to confirm this selection, the “Weiter” (Continue) button is clicked. After another delay, the ticket will be selected, and the text of the reservation extracted, based on the implementation shown in Figure 3. To verify the accuracy of the selected information, a message box is displayed, showing the relevant reservation text.



Figure 2: Text Scraping UiPath

The final step of the implementation, as depicted in Figure 4, focuses on file handling. This involves creating a new text file into which the extracted reservation details are added.

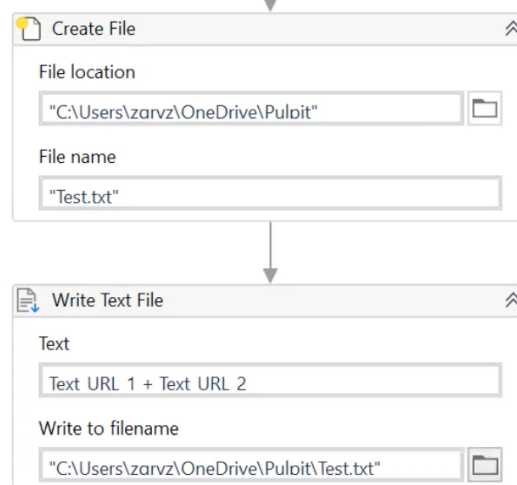


Figure 4: File Handling in UiPath