

**Anjalai Ammal Mahalingam Engineering College.**

**Department of Computer Science and Engineering**

Project 1 - Random User API

Submitted by,

E.PRIYASRI

(820422104061)

TABLE OF CONTENTS

* Problem Definition
* Abstract
* Introduction
* Proposed solution
* Tools and Techniques used
* Results and discussion

Project 1 - Random User API

## ****1. Problem Definition****

Creating diverse and realistic user profiles for testing applications is a common requirement in software development. Manual creation of such profiles is tedious, inefficient, and prone to errors. The goal is to automate the process using UiPath to fetch, process, and organize user data from the **Random User Generator API**. The data is categorized and stored in an Excel file for region-specific testing scenarios.

## ****2. Abstract****

This project leverages the power of UiPath to automate the creation and management of test user data, addressing the growing need for realistic and diverse user profiles in software development and testing. By integrating the **Random User Generator API**, the solution eliminates the manual effort of data generation, ensuring a streamlined and efficient workflow. The project revolves around retrieving user data in JSON format using UiPath's HTTP Request activity, which acts as the foundation for further data processing and organization.

The JSON data obtained from the API is deserialized into a structured format, allowing easy extraction of key attributes such as names, genders, email addresses, and geographic locations. Using UiPath’s data manipulation capabilities, these extracted fields are transformed into a DataTable format, which is highly versatile for handling and storing structured data.

The final output is saved into an Excel file, a universally accepted and accessible format for data storage and analysis. To enhance usability, the data is organized into multiple sheets, with each sheet dedicated to a specific country. This organization facilitates region-specific testing scenarios, allowing testers and developers to simulate and evaluate application behavior for different geographic demographics.

This automation not only simplifies the preparation of test data but also improves its quality and realism. By ensuring that the generated profiles reflect diverse and real-world scenarios, the solution supports more comprehensive testing of applications. It addresses challenges such as ensuring coverage of edge cases, validating location-based functionalities, and optimizing workflows dependent on dynamic user inputs.

Moreover, the project highlights the capability of UiPath to integrate with external APIs, showcasing how RPA tools can be used for advanced data manipulation tasks. It demonstrates the versatility of UiPath for automating repetitive and labor-intensive tasks, freeing up valuable resources for more strategic aspects of application development.

In conclusion, this project offers an efficient, scalable, and reliable method for generating and managing test user data. It underlines the potential of automation in bridging gaps in software testing and development, ultimately contributing to higher-quality outputs and reduced time-to-market for applications. This solution is not only practical but also adaptable, making it a valuable asset for various industries and testing environments.

## ****3. Introduction****

### ****3.1 Background****

Testing applications like social media platforms requires realistic user data. Using automation tools like UiPath for data generation ensures faster, more reliable processes than manual approaches.

### ****3.2 Objective****

To create a UiPath workflow that:

1. Retrieves data from the **Random User API**.
2. Processes JSON responses.
3. Organizes data into Excel sheets, categorized by user country.

### ****3.3 Relevance****

* Reduces manual effort in test data creation.
* Ensures accurate and diverse test profiles.
* Enhances scalability for testing across various scenarios.

## ****Proposed Solution****

### ****4.1 Workflow Overview****

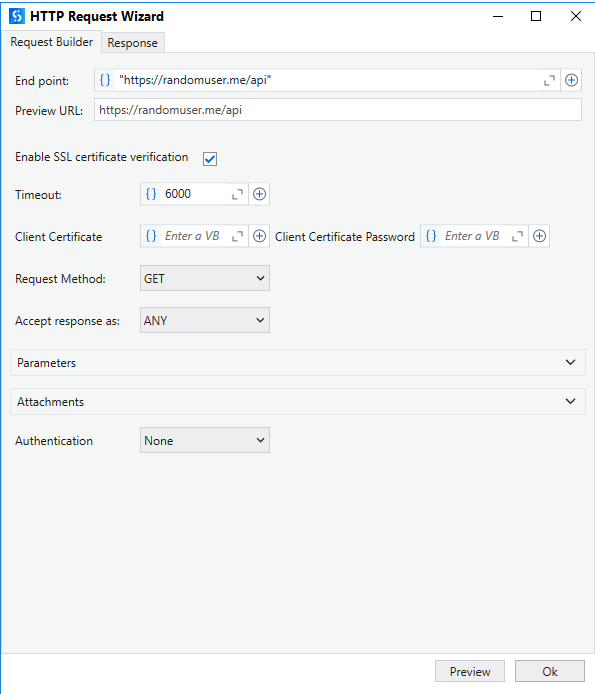
The UiPath workflow consists of the following steps:

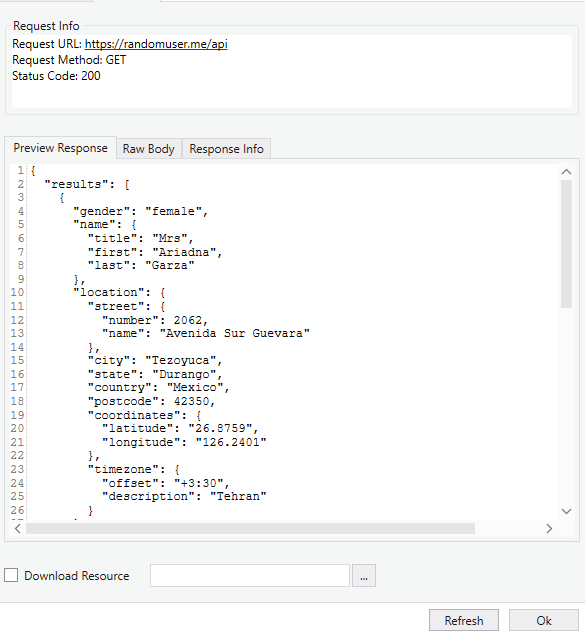
1. **HTTP Request Activity**: Fetch data from the Random User API.
2. **Deserialize JSON Activity**: Parse the API response.
3. **Data Extraction**: Extract relevant fields such as name, email, country, etc.
4. **DataTable Conversion**: Store extracted data into a DataTable.
5. **Excel Automation**: Save the data in Excel, creating separate sheets for each country.

### ****4.2 Detailed Steps****

#### **Step 1: API Request**

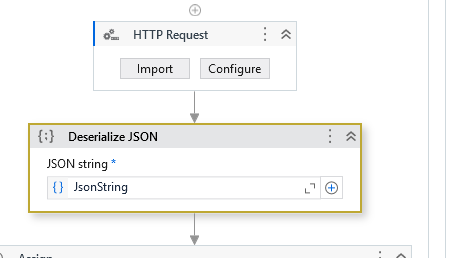
* Use the **HTTP Request** activity to send a GET request to the API:  
  **URL**: <https://randomuser.me/api/?results=100>
* Store the API response in a variable (e.g., apiResponse).





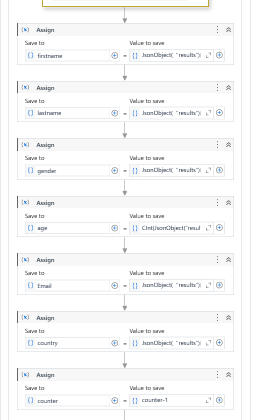
#### **Step 2: Deserialize JSON**

* Use the **Deserialize JSON** activity to parse apiResponse into a structured format.
* Assign the output to a variable (e.g., jsonObject).



#### **Step 3: Extract Data**

* Use a **For Each** activity to iterate through jsonObject("results").
* Extract fields like:
  + **First Name**: JsonObject(
  + "results")(0)("name")("first").ToString
  + **Last Name**: JsonObject( "results")(0)("name")("last").ToString
  + **Gender:** JsonObject( "results")(0)("gender").ToString
  + **Age:** CInt(JsonObject("results")(0)("dob")("age"))
  + **Email**: JsonObject( "results")(0)("email").ToString
  + **Country**: JsonObject( "results")(0)("location")("country").ToString

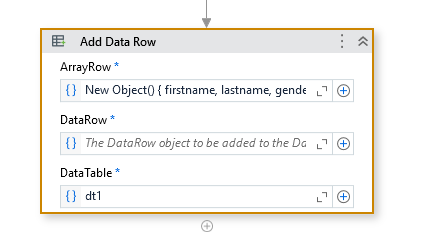


#### **Step 4: Create DataTable**

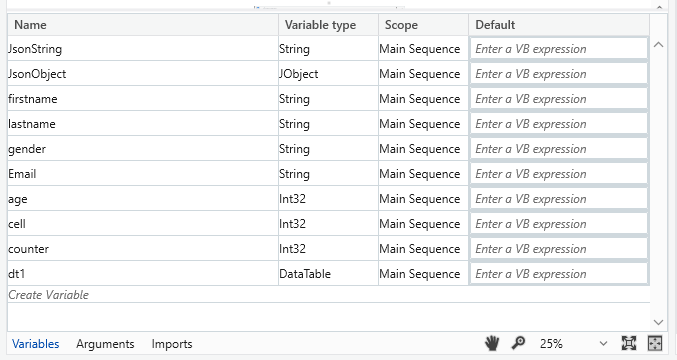
* Use the **Build DataTable** activity to define columns: First Name, Last Name, Email, Country, etc.



* Use the **Add Data Row** activity to add extracted data to the DataTable.

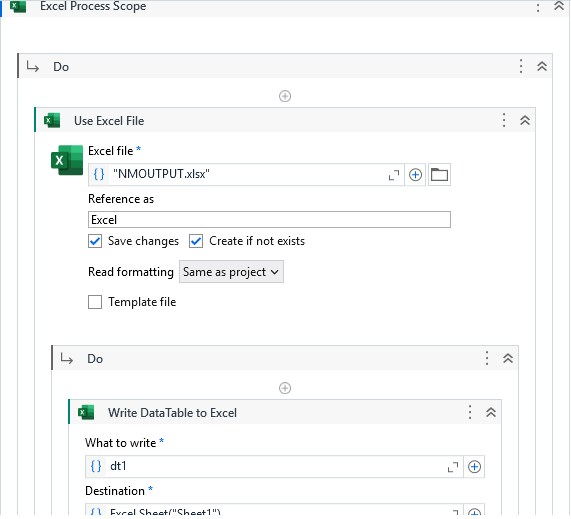


Variables used:



**Step 5: Save Data to Excel**

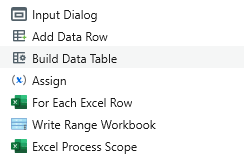
* Use the **Excel Application Scope** activity to open an Excel file.
* Use the **Write Range** activity to save the DataTable.
* Use a **Filter DataTable** activity to group data by country, creating separate sheets for each.

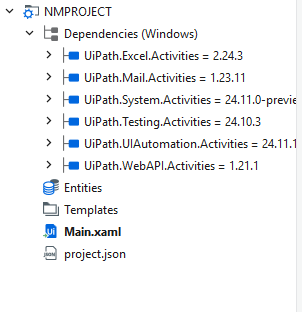


## ****5. Tools and Techniques Used****

### ****5.1 Tools****

1. **UiPath Studio**: For designing and implementing the automation workflow.
2. **Excel Application Scope**: To manage Excel files.
3. **HTTP Request Activity**: To fetch data from the API.
4. **Deserialize JSON Activity**: To parse JSON responses.





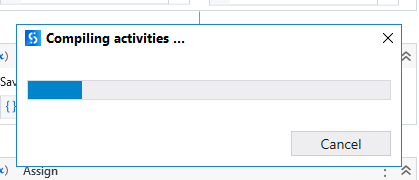
### ****5.2 Techniques****

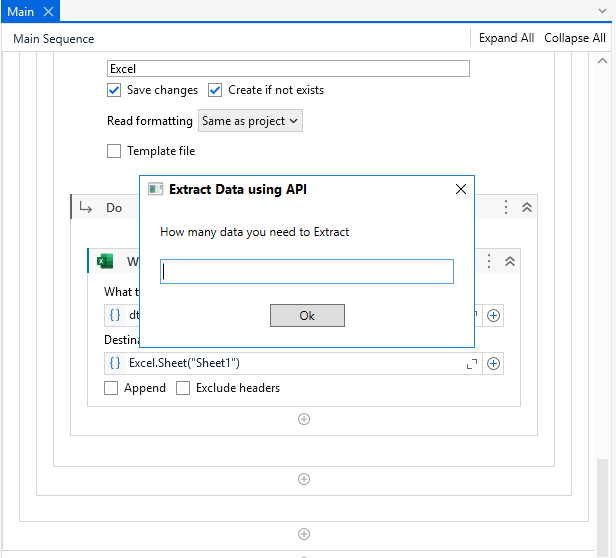
1. **API Integration**: UiPath's HTTP Request activity enables seamless interaction with web APIs.
2. **Data Manipulation**: The extracted JSON data is structured into DataTables.
3. **Excel Automation**: UiPath manages Excel files for storing and organizing data.

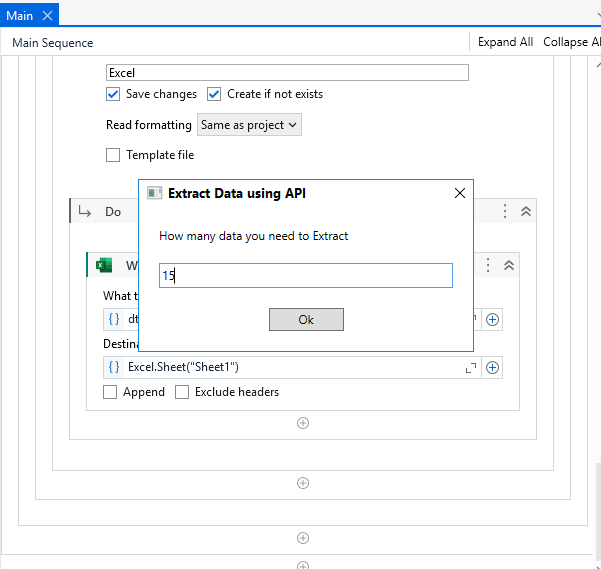
## ****6. Results and Discussion****

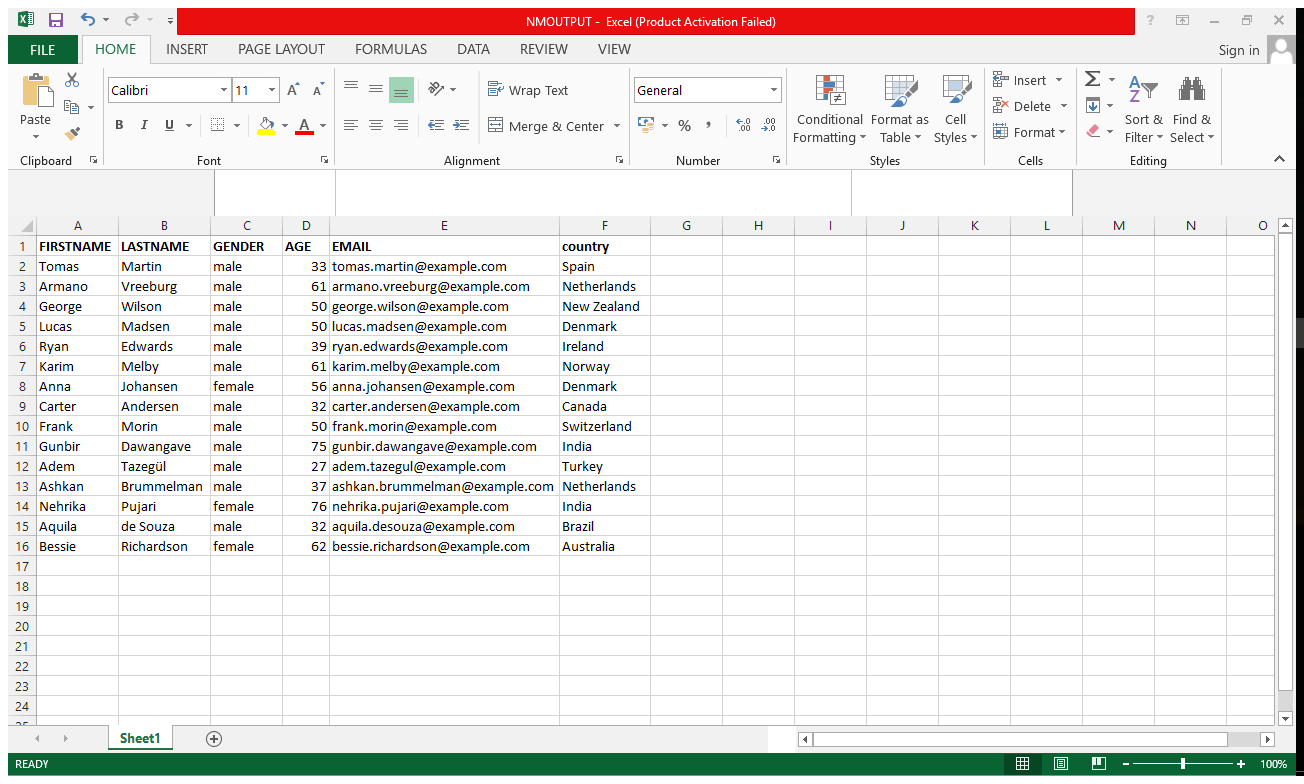
### ****6.1 Results****

* **Automated Data Fetching**: Successfully fetched 100 user profiles using the API.
* **Data Organization**: Processed and stored user data in Excel sheets categorized by country.









### ****6.3 Discussion****

The UiPath workflow demonstrated significant improvements in efficiency and accuracy compared to manual methods. Categorizing the data by country further enhanced its usability for region-specific testing.

## ****7. UiPath Workflow Design****

### ****7.1 Workflow Diagram****

1. Fetch API Data → Deserialize JSON → Extract Data → Build DataTable → Write to Excel.

## ****8. Challenges Faced****

1. **API Rate Limits**: Adjusted requests to avoid exceeding API limits.
2. **Data Mapping**: Ensured proper mapping of JSON fields to DataTable columns.

## ****9. Future Enhancements****

### ****Future Enhancements****

1. **Dynamic API Requests**  
   Enhancing the workflow to support dynamic API requests will provide users with the flexibility to customize the number of profiles fetched based on their specific testing needs. This can be achieved by introducing input parameters in the UiPath workflow, where users can specify the desired count of profiles. The parameter can be appended to the API URL as a query string, for example:

**URL**: <https://randomuser.me/api/?results=100>

This customization ensures scalability and caters to various testing scenarios, ranging from small-scale functional testing to large-scale load testing.

1. **Data Visualization**  
   Incorporating data visualization features can significantly enhance the analysis of user data. By utilizing tools such as Microsoft Excel charts or integrating with visualization platforms like Power BI or Tableau, users can generate graphical representations of the data. For instance, pie charts can show the distribution of users by gender or country, while bar graphs can illustrate age group demographics. Visual insights make it easier to spot trends, identify patterns, and validate the realism of the generated data.
2. **Integration with Databases**  
   To improve scalability and data management, the workflow can be extended to store user profiles in a relational database instead of an Excel file. By integrating with database systems such as SQL Server, MySQL, or PostgreSQL, the solution can handle larger datasets efficiently. Database integration also enables advanced querying, filtering, and reporting capabilities. Additionally, connecting the workflow to a cloud-based database allows for centralized data storage, facilitating collaboration among team members across different locations.

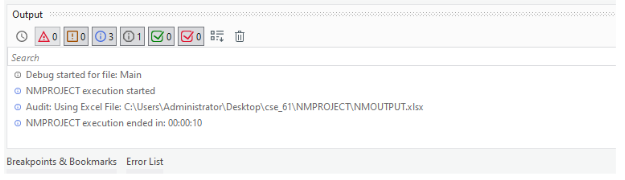
These enhancements will make the project more robust, user-friendly, and adaptable to advanced testing environments.

## ****10. Conclusion****

Using UiPath, this project effectively automated the end-to-end process of generating, processing, and organizing realistic user profiles. By leveraging the **Random User Generator API**, the workflow retrieves user data dynamically and structures it into a format ready for testing and analysis. This approach eliminates the manual effort involved in creating diverse datasets, reducing time and labor while ensuring the accuracy and diversity of the data.

The solution is designed to handle large volumes of data efficiently, making it highly scalable for various application testing requirements. Whether a project demands a few profiles for functional testing or thousands for performance testing, the workflow can be adjusted to meet the needs without additional development effort. The categorization of user data by region, stored in separate Excel sheets, enhances usability, especially for applications requiring geographic-specific testing.

This adaptability allows the workflow to integrate seamlessly with other automation processes or tools, ensuring it fits into broader testing frameworks. As a result, this project not only streamlines the creation of test data but also improves its quality and relevance, enabling more comprehensive testing scenarios. Ultimately, the solution showcases the power of UiPath in automating complex, repetitive tasks while maintaining scalability, efficiency, and flexibility.



FOR VIEW PROJECT REFER THE BELOW DRIVE LINK:

<https://drive.google.com/drive/folders/16WaOULr0gqC1Gv6LENAqNCz1Ktj5oBTa?usp=sharing>