Automation Testing

Data-Driven Testing

A Day in the Life of an Automation Testing Engineer

John has been assigned a task to perform data-driven testing in Postman. He is now exploring the capabilities of Postman with data-driven testing.

He plans data-driven testing using CSV and JSON files as he must carry out a test with numerous sets of data. The files will allow him to add or change data fields by only updating the data files instead of conducting a Test script to update the data.

After completing this lesson, John can create and run data-driven API testing in Postman.

Learning Objectives

By the end of this lesson, you will be able to:

- Identify data-driven testing
- List different forms of data-driven testing
- Perform data-driven testing from Postman

FULL STACK

Data-Driven Testing

simplilearn

Data-Driven Testing

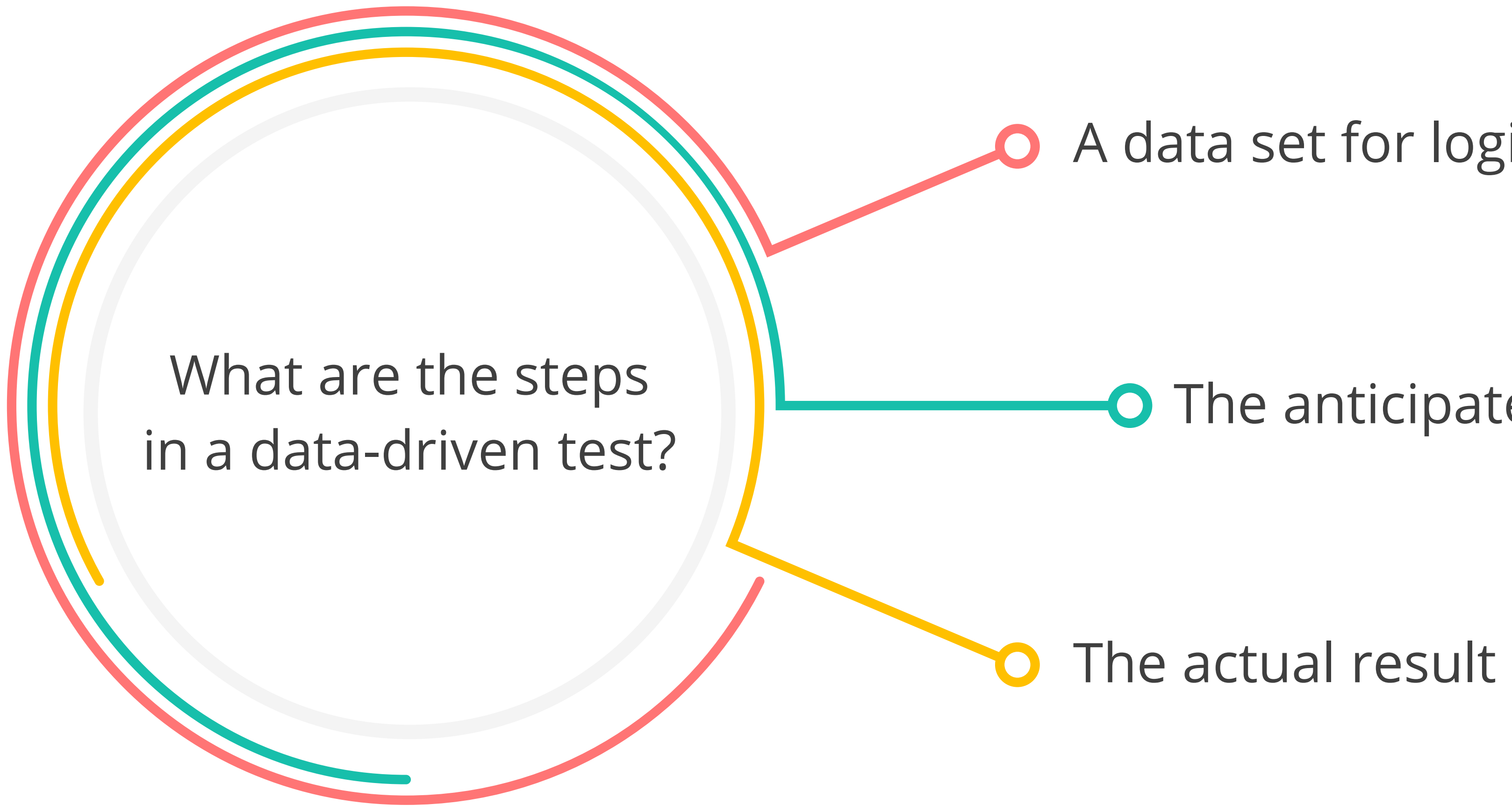
Typically, an automated data-driven test consists of the following elements:

A data source, such as a text file, file directory, data pool, JDBC source, or spreadsheet delimited by commas

Test steps, where users can use the information in additional steps like requests, assertions, XPath queries, or scripts

A data loop, where users can continue the test with the subsequent rows of data

Data-Driven Testing


What are the steps in a data-driven test?
A data set for logic (request or reply model) testing
The anticipated result
The actual result

Testers frequently utilize a table to conduct the test, separating the data from the test commands and comparing the predicted and actual outputs.

Data-Driven Testing Sequence

What entails a data-driven test sequence?

- Data from storage is retrieved in parts
- The data is injected into API requests based on tokenization criteria
- The outcomes are confirmed
- The test can be repeated with the next row of the input data

This pattern will persist until the test exhausts the entire data store or meets an error, depending on how the user configures the test.

Benefits a Data-Driven API Test

How to construct a data-driven API test?

- Make a list of the predetermined series of operations that an application must carry out with the API
- Gather testable data and store it in a table or spreadsheet, both considered a form of storage
- Create a test logic with a predetermined set of test data
- Substitute a set of variable test data for the fixed test data
- Provide the value that is stored in the data storage to the variables

Benefits of Data-Driven Testing

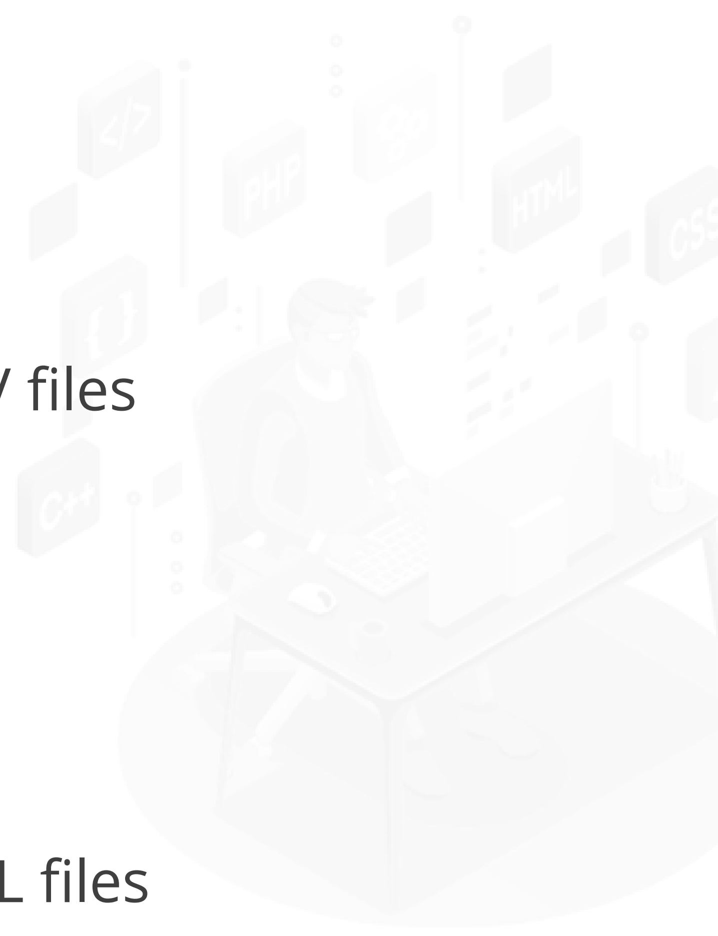
The benefits of data-driven automated testing are:

- **Reduced human error:** Unlike manual testing, test data is read from files instead of being entered manually.
- **Maintainability:** As test data and test scripts are stored independently and not dependent on one another, updating the test scripts or the test data is hassle-free.
- **Broader test coverage:** Users can get excellent test coverage by using high-quality test data that fully replicates the production data.
- **Time-saving:** The automated test run is scheduled every night when no one is working. Significant time is saved in the testing procedures because the results are verified in the morning.

Test Data Storage

Test data for Data-Driven Testing is stored in any of the following file formats:

- Text files
- Arrays
- Database tables
- Excel files
- CSV files
- XML files



Data-Driven Automated Testing Tools and Frameworks

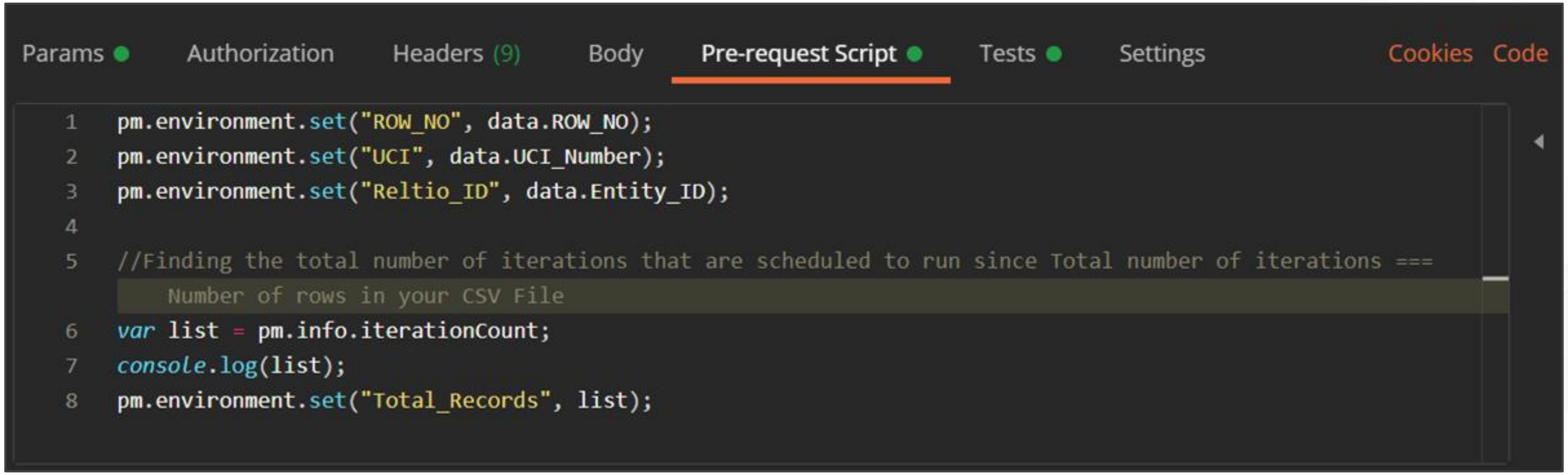
Some data-driven automated testing tools and frameworks are:

- Apache POI, TestNG, and Selenium (for reading and writing in Excel sheet)
- Testsigma
- QTP
- Katalon Studio
- TestComplete

Postman for Data-Driven Testing

Postman for Data-Driven Testing

In data-driven testing, users perform multiple test runs with different data variables.



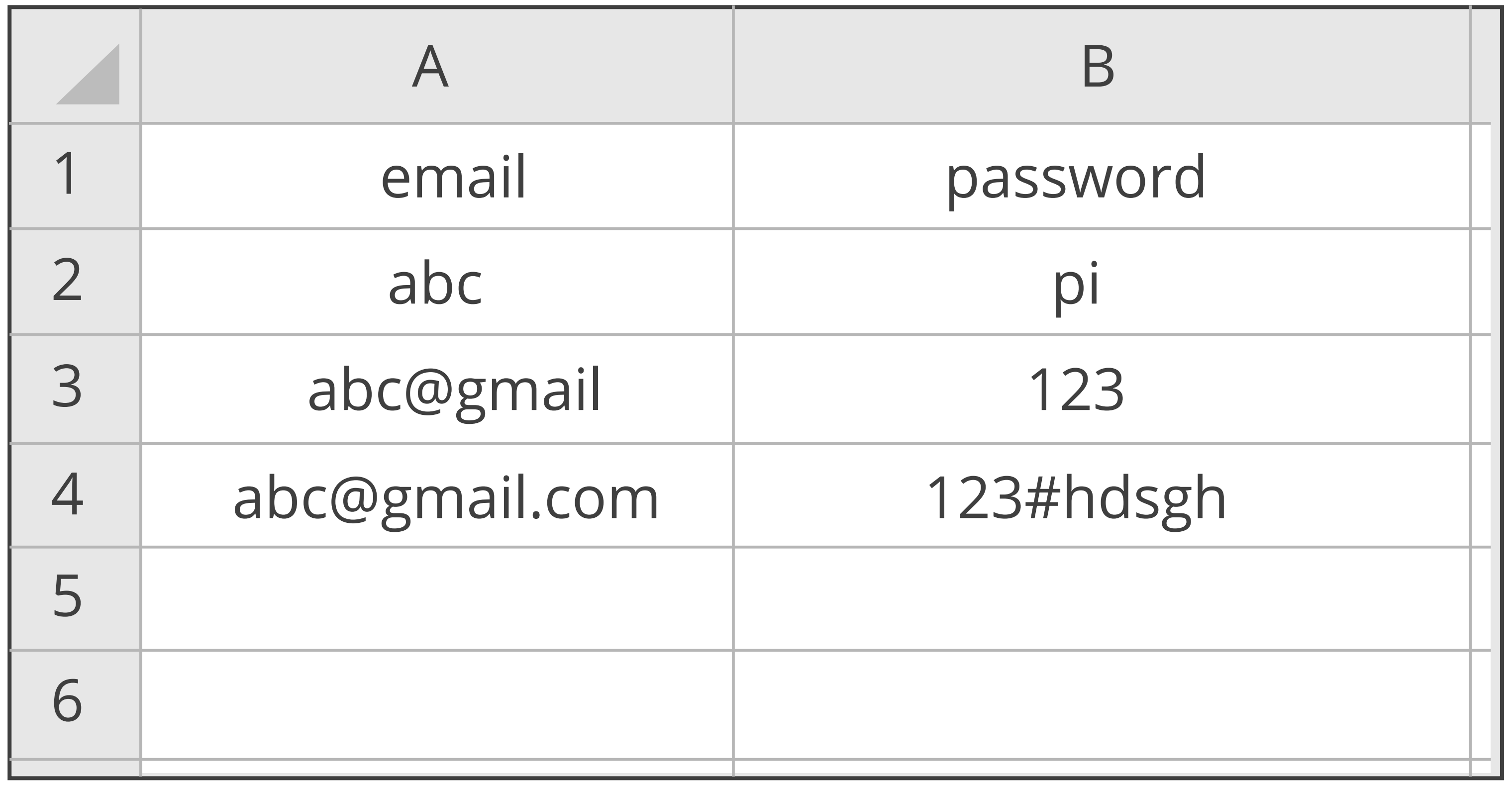
```
pm.environment.set("ROW_NO", data.ROW_NO);
pm.environment.set("UCI", data.UCI_Number);
pm.environment.set("Reltio_ID", data.Entity_ID);

//Finding the total number of iterations that are scheduled to run since Total number of iterations ===
    Number of rows in your CSV File
var list = pm.info.iterationCount;
console.log(list);
pm.environment.set("Total_Records", list);
```

Postman-Supported File Types

Postman offers to read data from external sources, such as CSV or JSON, which helps achieve data-driven API automation testing.

- Using CSV file



	A	B
1	email	password
2	abc	pi
3	abc@gmail	123
4	abc@gmail.com	123#hdsgh
5		
6		

- Using JSON file

```
[
        {
                "email" : "abc",
                "password" : "pi"
},      {
                "email" : "abc@gmail",
                "password" : 123
},      {
                "email" : "abc@gmail.com",
                "password" : "123#hdsgh"
        }
]
```

Key Takeaways

- Users can perform data-driven automation testing using Postman.
- Two files are used in a data-driven approach: CSV and JSON.
- It is possible to execute a Collection on runner to fetch a variable from a CSV file and a JSON file.