Cucumber - Data Driven Testing

In Cucumber, `Scenario Outline` is used when you want to **run the same scenario multiple times with different sets of input data**. You define a template using placeholders, and then provide multiple data rows using the `Examples` keyword.

✅ Syntax of `Scenario Outline` and `Examples`

```gherkin
Feature: Login Feature

  Scenario Outline: Successful login with valid credentials
    Given user is on the login page
    When user enters username "<username>" and password "<password>"
    Then user should see the dashboard

  Examples:
    | username  | password  |
    | user1     | pass123   |
    | admin     | admin123  |
```

🔄 Explanation

- `Scenario Outline`: Defines a reusable test scenario.
- `<username>` and `<password>` are **placeholders**.
- `Examples` table provides **actual values** for the placeholders.
- The scenario will run **once per row** in the `Examples` table.

✅ Step Definition in Java

If you're using Java with Cucumber:

```java
@When("user enters username {string} and password {string}")
public void user_enters_username_and_password(String username, String password) {
    System.out.println("Username: " + username);
    System.out.println("Password: " + password);
    // Code to enter username and password in your app
}
```

🧪 Output

It will execute 2 test cases:

1. With username = `user1` and password = `pass123`
2. With username = `admin` and password = `admin123`

✅ Use Case Example

```gherkin

Scenario Outline: Search results should show related items
  Given user is on the home page
  When user searches for "<searchTerm>"
  Then search results should include "<searchTerm>"

  Examples:
    | searchTerm  |
    | laptops     |
    | mobiles     |
    | headphones  |
```

Would you like a **mini project or real-time test case** using `Scenario Outline` for Selenium or API testing?

In Cucumber, **Data Tables** are used to pass multiple rows of data to a step, usually when you want to handle **structured input** like lists or key-value pairs **within a single scenario** (not like `Scenario Outline`, which repeats scenarios).

✅ Syntax: Data Table in Feature File

🔹 Example 1: Key-Value Data Table (Map)

```gherkin
Feature: Login

  Scenario: Login with valid credentials
    Given the user logs in with the following credentials:
      | username | admin   |
      | password | admin123 |
```

🔹 Java Step Definition (Using Map)

```java
@Given("the user logs in with the following credentials:")
public void the_user_logs_in_with_the_following_credentials(io.cucumber.datatable.DataTable
dataTable) {
    Map<String, String> credentials = dataTable.asMap(String.class, String.class);
    String username = credentials.get("username");
    String password = credentials.get("password");

    System.out.println("Username: " + username);
    System.out.println("Password: " + password);

    // Use these credentials in your login function
}
```

🔹 Example 2: List of Data Rows (List of Lists)

```gherkin
Scenario: Add multiple products to cart
  Given the user adds the following products to the cart:
    | Product Name | Quantity |
    | Laptop       | 1        |
    | Mouse        | 2        |
    | Keyboard     | 1        |
```

🔹 Java Step Definition (List of Lists)

```java
@Given("the user adds the following products to the cart:")
public void the_user_adds_the_following_products_to_the_cart(io.cucumber.datatable.DataTable
dataTable) {
    List<List<String>> rows = dataTable.asLists(String.class);

    for (int i = 1; i < rows.size(); i++) { // skip header row
        String product = rows.get(i).get(0);
        int quantity = Integer.parseInt(rows.get(i).get(1));
```

```
        System.out.println("Adding product: " + product + " with quantity: " + quantity);
        // call addToCart(product, quantity)
    }
}
```

✅ When to Use Data Tables

Use Case	Use Scenario Outline?	Use Data Table?
Multiple sets of test data	✅ Yes	🚫 No
Structured input for a single step	🚫 No	✅ Yes
Input like form fields (key-value pairs)	🚫 No	✅ Yes

🔄 Would you like a real-time login page test using Data Tables in Selenium?

Using **Lists** and **Maps** in Java for test data is a great way to manage inputs dynamically and cleanly, especially in test automation with Cucumber, Selenium, or Rest Assured.

✅ 1. Using `List` in Java for Test Data

🔹 Use Case: List of usernames to test login

```
java

List<String> usernames = new ArrayList<>();
usernames.add("admin");
usernames.add("user1");
usernames.add("guest");

for (String username : usernames) {
    System.out.println("Testing login with: " + username);
    // login(username, defaultPassword);
}
```

✅ 2. Using `Map` in Java for Key-Value Test Data

🔹 Use Case: Single user credentials

```
java

Map<String, String> userCredentials = new HashMap<>();
userCredentials.put("username", "admin");
userCredentials.put("password", "admin123");

System.out.println("Username: " + userCredentials.get("username"));
System.out.println("Password: " + userCredentials.get("password"));

// login(userCredentials.get("username"), userCredentials.get("password"));
```

✅ 3. List of Maps (Table-Like Data)

🔹 Use Case: Multiple user login attempts

```
java

List<Map<String, String>> users = new ArrayList<>();

Map<String, String> user1 = new HashMap<>();
user1.put("username", "admin");
user1.put("password", "admin123");

Map<String, String> user2 = new HashMap<>();
user2.put("username", "guest");
user2.put("password", "guest123");

users.add(user1);
users.add(user2);

for (Map<String, String> user : users) {
    System.out.println("Testing login for: " + user.get("username"));
```

```
    // login(user.get("username"), user.get("password"));
}
```

✅ 4. Data from Cucumber Data Table into List or Map

If you're using this in a **Cucumber Step Definition**, you can automatically convert a Gherkin table into:

- **List of Maps:**

```
gherkin

Given the user tries to login with the following users:
  | username | password  |
  | admin    | admin123  |
  | guest    | guest123  |
```

Step Definition:

```
java

@Given("the user tries to login with the following users:")
public void login_with_multiple_users(DataTable dataTable) {
    List<Map<String, String>> users = dataTable.asMaps(String.class, String.class);

    for (Map<String, String> user : users) {
        String username = user.get("username");
        String password = user.get("password");
        System.out.println("Logging in with: " + username + " / " + password);
        // login(username, password);
    }
}
```

✅ When to Use What?

Scenario	Use
One item with multiple values (e.g., users)	`List<Map<>>`
Simple list of items (e.g., search terms)	`List<String>`
Key-value pairs (e.g., form input)	`Map<>`
Full data table from Cucumber	`DataTable.asMaps()` or `asLists()`

Would you like to integrate this into a **Selenium or API test framework**, or need a **mini project** example with this approach?