

Cucumber - Error handling and Debugging

Step definition mismatches in Cucumber typically occur when the steps written in the `.feature` file don't correctly match the regular expressions defined in your step definitions (in Java, Ruby, JavaScript, etc.). Here's how to identify and fix them step-by-step:

1. Understand the Error

When you run your Cucumber tests and see something like:

```
typescript

You can implement missing steps with the snippets below:
@Given("user is on login page")
public void user_is_on_login_page() {
    // Write code here
}
```

It means Cucumber **couldn't find a matching step definition** for the step in your `.feature` file.

2. Fixing the Mismatch – Step-by-Step

Step A: Match the Regex

Ensure your step definition's regular expression matches **exactly** what's written in the `.feature` file.

Example mismatch:

Feature file:

```
gherkin

Given user is on the login page
```

Step definition:

```
java

@Given("user is on login page")
public void user_is_on_login_page() {
    //...
}
```

Fix: Add "the" in the step definition regex:

```
java

@Given("user is on the login page")
```

Step B: Handle Parameters Correctly

Feature file:

```
gherkin

When user enters "admin" and "password123"
```

Mismatch Step definition:

```
java

@When("user enters admin and password123") // ❌ hardcoded
```

Fix using regex parameters:

```
java

@When("user enters {string} and {string}")
public void user_enters_credentials(String username, String password) {
    // Use username and password
}
```

✅ Step C: Remove Extra Whitespaces or Typos

Even one extra space or character will cause mismatches. Double-check the wording.

✅ Step D: Copy Suggestions from Console

Cucumber usually prints the missing step definition template in the terminal. You can copy that into your step definition class to avoid mismatch.

✅ Step E: Check for Ambiguous Steps

If you have **more than one** step definition that matches the same step, Cucumber throws an *ambiguous step definition* error.

Fix: Ensure only **one** step definition matches the step. You might need to make your regex more specific.

✅ Step F: Step Definition Binding Issues

- Make sure your step definition class is in the correct package.
- Ensure it's being picked up by the test runner (i.e., it's included in the **glue** option of your Cucumber runner).

```
java

@CucumberOptions(
    features = "src/test/resources/features",
    glue = "stepDefinitions" // ✅ This should match the package of your step definitions
)
```

Bonus Tips

- Use `{int}`, `{string}`, `{word}` in step definitions for flexibility.
- Use `^` and `$` to anchor the regex if needed.
- Organize your step definitions clearly to avoid accidental overlaps.

Example Fix Summary

Feature File:

```
gherkin

Given user is on the login page
When user enters "admin" and "admin123"
Then user should be redirected to the dashboard
```

Matching Step Definitions (Java):

```
java

@Given("user is on the login page")
public void user_is_on_the_login_page() { }

@When("user enters {string} and {string}")
public void user_enters_credentials(String username, String password) { }

@Then("user should be redirected to the dashboard")
public void user_should_be_redirected_to_dashboard() { }
```

Let me know if you'd like to **debug a specific mismatch** — you can paste the step and definition here.

Cucumber Duplicate Steps Error – Explained and Fixed

When you see a **duplicate or ambiguous step definition error**, it means that **Cucumber found more than one step definition that matches a step in your feature file**. This creates confusion, as it doesn't know which method to run.

✓ 1. Example of Duplicate Step Issue

Feature File:

```
gherkin

Given user is on the login page
```

✗ Two Step Definitions:

```
java

@Given("user is on the login page")
public void stepOne() {
    // implementation A
}

@Given("user is on the login page")
public void stepTwo() {
    // implementation B
}
```

Error Message:

```
vbnet

AmbiguousStepDefinitionsException: 'Given user is on the login page' matches more than one step definition.
```

✓ 2. How to Fix It

✓ A. Remove the Duplicate

Only **one method** should match a given step.

Fix:

Keep just one step definition:

```
java

@Given("user is on the login page")
public void stepOne() {
    // single implementation
}
```

✓ B. Make the Regex More Specific

Sometimes you have two similar steps, and you accidentally wrote a generic regex that matches both.

Example:

```
java

@Given("user is on {string}")
public void onPage(String page) { }

@Given("user is on the login page")
public void onLoginPage() { }
```

When the feature says:

```
gherkin

Given user is on the login page
```

Both step definitions match.

Fix: Make one regex more specific or remove overlap:

```
java

// Use only this
@Given("user is on the {word} page")
public void onPage(String page) { }
```

Or remove one if it's unnecessary.

✓ C. Ensure Clean Glue Paths

If you accidentally define the same step in **two different packages** and include both in your **glue**, it may lead to duplication.

Fix: Keep glue clean in your runner:

```
java

@CucumberOptions(
    features = "src/test/resources/features",
    glue = {"stepDefinitions"} // ✓ Keep only relevant package
)
```

✓ D. Avoid Copy-Paste Duplication

Sometimes duplicate steps are created while copying code for new test cases. Always search before writing a new step.

🔪 Best Practices to Avoid Duplicate Steps

1. **Use Parameterized Steps** like `{string}`, `{int}` to avoid writing similar steps multiple times.
 2. **Organize Step Definitions** logically: one class per feature or topic.
 3. **Use search** (`Ctrl+Shift+F` or `grep`) before adding new steps.
 4. **Avoid overly generic regex** like `@Given(".*")`.
-

Need Help Finding Duplicates?

If you share your feature file + step definition code (or their structure), I can help pinpoint and resolve the duplication directly.

Let me know!

🚫 Undefined Steps in Cucumber – Causes & How to Fix

If you're seeing “**undefined step**” errors in Cucumber, it means Cucumber **can't find a step definition that matches the step written in your .feature file**.

🧠 Common Error Message

```
csharp
```

```
Undefined step: Given user is on the login page
You can implement missing steps with the snippets below:
@Given("user is on the login page")
public void user_is_on_the_login_page() {
    // Write code here
}
```

🔧 Step-by-Step Fix for Undefined Steps

✅ 1. Check Step Text Match

Feature:

```
gherkin
```

```
Given user is on the login page
```

Incorrect Step Definition:

```
java
```

```
@Given("user is on login page") // ❌ Missing "the"
```

Fix:

```
java
```

```
@Given("user is on the login page") // ✅ Must match exactly
```

✅ 2. Use {string}, {int} for Dynamic Values

Feature:

```
gherkin
```

```
When user logs in with "admin" and "pass123"
```

Missing Step:

```
java

@When("user logs in with admin and pass123") // ❌ hardcoded
```

Fix:

```
java

@When("user logs in with {string} and {string}")
public void login(String username, String password) {
    // use username, password
}
```

✅ 3. Check if Step Definitions Are in the Right Package

Your step definition class **must be in the package(s) listed in glue** in the test runner.

Fix:

In `TestRunner.java`:

```
java

@cucumberOptions(
    features = "src/test/resources/features",
    glue = {"stepDefinitions"} // ✅ Must match your step definition package
)
```

✅ 4. Make Sure Step Definitions Are Compiled

If you're using Maven or Gradle, sometimes the step definitions may not have compiled.

Fix:

- Rebuild your project:

```
bash

mvn clean install
```

- Or reload project in IntelliJ / Eclipse.

✅ 5. Copy Missing Step from Console

When you run the test, Cucumber shows suggestions:

```
typescript

@Then("user is on dashboard page")
public void user_is_on_dashboard_page() {
    // Write code here
}
```


Tip: Copy and paste these to your step definition class.

✓ 6. Avoid Typos and Extra Spaces

Even a small typo or extra whitespace can break matching.

✓ Example Working Setup

Feature File:

```
gherkin

Feature: Login

Scenario: Valid login
  Given user is on the login page
  When user enters "admin" and "admin123"
  Then user should be redirected to the dashboard
```

Step Definitions (Java):

```
java

@Given("user is on the login page")
public void user_on_login_page() { }

@When("user enters {string} and {string}")
public void enter_credentials(String username, String password) { }

@Then("user should be redirected to the dashboard")
public void redirect_to_dashboard() { }
```

✓ Summary

Cause	Fix
Step text mismatch	Match feature step exactly
Hardcoded values	Use {string} , {int}
Wrong package or no glue	Add correct package to glue in runner
Not compiled	Build project / reload IDE
Typos or extra spaces	Review step carefully

If you'd like, paste your feature + step definition + runner file and I'll help you debug it directly.

⚠️ "Skipped steps due to previous failures" in Cucumber – Explained & Fixed

This is **not an error with your step definitions**, but a symptom of an earlier step **failing** (either through an exception, assertion, or undefined step). Cucumber **skips all subsequent steps in the scenario** once a failure happens.

🧠 Why This Happens

Cucumber follows **Gherkin's fail-fast** approach:

❌ If one step fails, 🚫 all remaining steps in that scenario are skipped.

🔧 Example:

🟡 Feature:

```
gherkin

Scenario: User login
  Given user is on the login page
  When user enters "admin" and "wrongpass"
  Then user should be redirected to the dashboard
```

🔴 Test Output:

```
pgsql

Given user is on the login page ✓
When user enters "admin" and "wrongpass" ❌ FAILED (e.g., AssertionError)
Then user should be redirected to the dashboard ⚠️ SKIPPED
```

✅ How to Fix It

✅ 1. Look at the First Failure

Check the console/logs for the **exact reason** of the **first failure** (usually the step before the first skipped step).

Common causes:

- `NullPointerException`
- `AssertionError`
- `NoSuchElementException` (for Selenium)
- Undefined step (missing implementation)

✅ 2. Fix the Root Cause

Examples:

- ❌ Assertion failed:

```
java
Assert.assertEquals(actualTitle, "Expected Title"); // Wrong title
```

- ❌ Element not found:

```
java
driver.findElement(By.id("nonexistent-id")).click(); // No such element
```

- ❌ Step undefined:

```
gherkin
Given user navigates to the settings page
```

But no matching @Given step → causes all next steps to be skipped

✅ 3. Re-run After Fixing



Once the **real cause is fixed**, re-run the test. Skipped steps should now execute normally.

✅ Optional: Use Hooks to Handle Failures Gracefully

You can use @AfterStep or @After hooks to log failure reasons or take screenshots:

```
java
@AfterStep
public void afterStep(Scenario scenario) {
    if (scenario.isFailed()) {
        System.out.println("Step failed: " + scenario.getName());
        // Take screenshot if using Selenium
    }
}
```

✅ Summary Table

 Cause	 Fix
Previous step failed	Check logs, fix assertion or exception
Step not implemented	Implement the missing step
Selenium element not found	Fix locator / wait properly
Incorrect logic	Fix test or app code logic

If you share the **failing step** or stack trace, I can help you pinpoint the failure and fix it step-by-step.