

Scenario 2 – Fail – Update User: username field is not supported in the update API's request body. As a result, attempting to update the username will lead to a failed update or an unrecognized field in the response. Please find the code , report and console output below.

Code:

```
package com.api;

import com.microsoft.playwright.*;
import com.microsoft.playwright.options.RequestOptions;
import org.json.JSONObject;

import java.util.Map;

public class PlaywrightAutomationAPI {

    public static void main(String[] args) {
        try (Playwright playwright = Playwright.create()) {

            // Initialize API context with base URL and headers
            APIRequestContext request = playwright.request().newContext(
                new APIRequest.NewContextOptions()
                    .setBaseURL("https://reqres.in")
                    .setExtraHTTPHeaders(Map.of(
                        "x-api-key", "reqres-free-v1",
                        "Content-Type", "application/json"
                    )));
    }

    // Step 1: Create a user
}
```

```
System.out.println("== Step 1: Create a user ==");

String createPayload = """
{
  "email": "george.bluth@reqres.in",
  "password": "1234"
}
""";

APIResponse createResponse = request.post("/api/register",
  RequestOptions.create().setData(createPayload));

int createStatus = createResponse.status();

System.out.println("Create User - HTTP Status Code: " + createStatus);

String userId = null;

if (createStatus == 200) {
  String createBody = createResponse.text();
  System.out.println("Create User - Response Body:\n" + createBody);

  JSONObject json = new JSONObject(createBody);
  String token = json.getString("token");
  System.out.println("User created successfully. Token: " + token);
}

// Fetch user list to get an existing userId (since /register doesn't return one)
APIResponse listResponse = request.get("/api/users?page=1");
JSONObject listJson = new JSONObject(listResponse.text());
```

```
userId =  
String.valueOf(listJson.getJSONArray("data").getJSONObject(0).getInt("id"));  
  
System.out.println("Fetched User ID from list: " + userId);  
  
} else {  
  
    System.out.println("User creation failed.");  
  
    return;  
  
}
```

```
// Step 2: Get the created user details  
  
System.out.println("\n==== Step 2: Get the created user details ===");  
  
APIResponse getResponse = request.get("/api/users/" + userId);  
  
int getStatus = getResponse.status();  
  
System.out.println("Get User - HTTP Status Code: " + getStatus);  
  
  
if (getStatus == 200) {  
  
    String getBody = getResponse.text();  
  
    System.out.println("Get User - Response Body:\n" + getBody);  
  
  
    JSONObject userJson = new JSONObject(getBody).getJSONObject("data");  
  
    System.out.println("User Name: " + userJson.getString("first_name") + " " +  
userJson.getString("last_name"));  
  
    System.out.println("User Email: " + userJson.getString("email"));  
  
} else {  
  
    System.out.println("Failed to fetch user details.");  
  
    return;  
  
}
```

```
// Step 3: PUT request with NO payload  
  
System.out.println("\n==== Step 3: PUT request without payload ===");
```

```
APIResponse updateResponse = request.put("/api/users/" + userId,
    RequestOptions.create()
        .setHeader("x-api-key", "reqres-free-v1")
        .setHeader("Content-Type", "application/json")
);
int updateStatus = updateResponse.status();
System.out.println("Update User - HTTP Status Code: " + updateStatus);

String updateBody = updateResponse.text();
System.out.println("Update User - Response Body:\n" + updateBody);

if (updateStatus == 200) {
    try {
        JSONObject updateJson = new JSONObject(updateBody);
        if (updateJson.has("updatedAt")) {
            System.out.println("User updated successfully at: " +
updateJson.getString("updatedAt"));
        } else {
            System.out.println("Update succeeded, but 'updatedAt' not found in response.");
        }
    } catch (Exception e) {
        System.out.println("Response is not valid JSON or missing fields.");
    }
} else {
    System.out.println("Failed to update user.");
}
```

```

} catch (Exception e) {

    e.printStackTrace();

}

}

}

```

Extent Report:

Please click report below to check failed scenario:



PlaywrightApiReport.html

Test Step	Status	Timestamp	Details
Step 1: Create a user and validate response	Pass	11.03.2025 2:38:10 PM	Update User - HTTP Status Code: 200
Step 2: Get the created user details and validate	Pass	11.03.2025 2:38:11 PM	Update User - Response Body: {"name": "Sanskruti", "updatedAt": "2025-11-03T12:17:54.416Z"}
Step 3: Update user's name and validate	Fail	11.03.2025 2:38:11 PM	Failed to update user.

Console output:

UPDATE - PASS:

```

==> Step 3: Update user's name and validate ==
Update User - HTTP Status Code: 200
Update User - Response Body:
{"name": "Sanskruti", "updatedAt": "2025-11-03T12:17:54.416Z"}
User updated successfully at: 2025-11-03T12:17:54.416Z

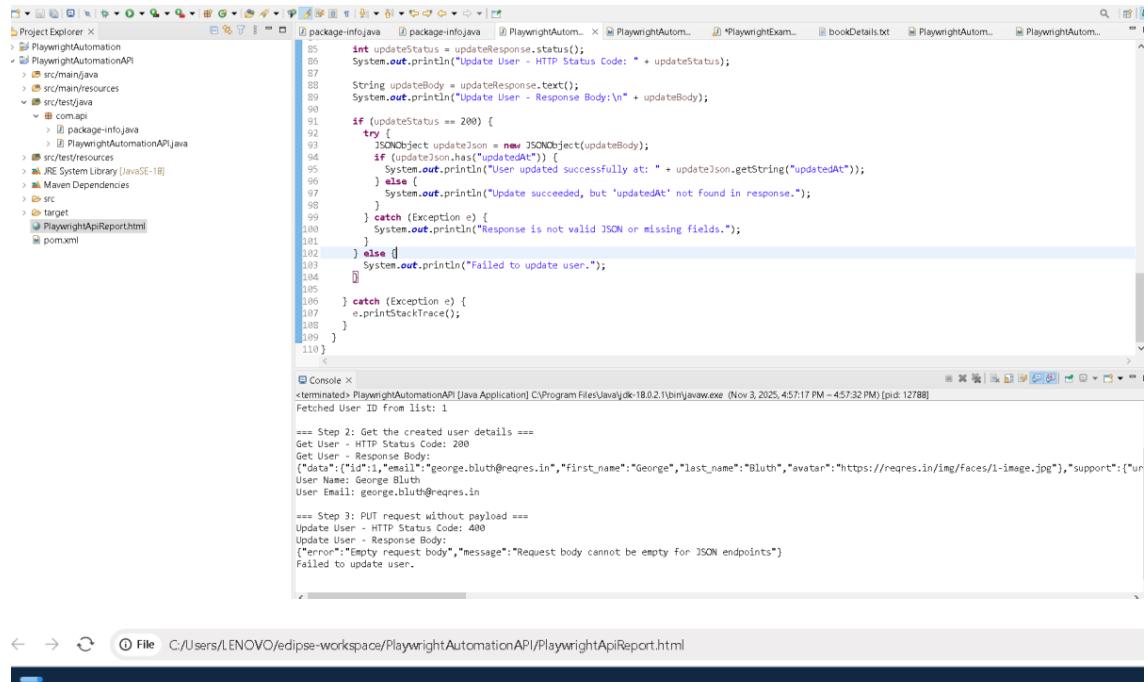
```

UPDATE-FAIL

```

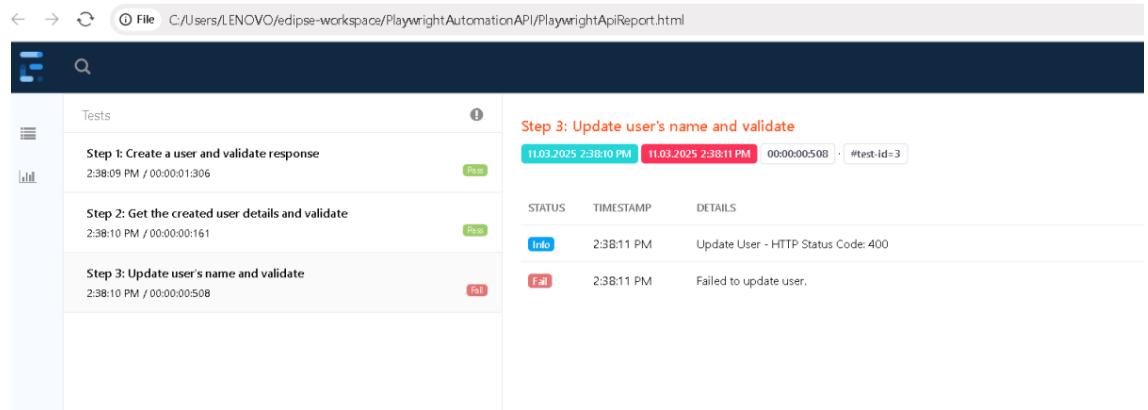
==> Step 3: PUT request without payload ==
Update User - HTTP Status Code: 400
Update User - Response Body:
{"error": "Empty request body", "message": "Request body cannot be empty for JSON endpoints"}
Failed to update user.

```



The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure for "PlaywrightAutomation".
- Code Editor:** Displays Java code for updating a user. The code uses the `updateStatus` method from the `package-info.java` file. It handles a successful update (status 200) by printing the updated JSON object and its `updatedAt` field. If the update fails (status != 200), it prints an error message.
- Console:** Shows the output of the application. It prints "Fetched User ID from list: 1". Then it performs three steps:
 - Step 2: Gets user details (status 200). Response body: {"data": [{"id": 1, "email": "george.bluth@reqres.in", "first_name": "George", "last_name": "Bluth", "avatar": "https://reqres.in/img/faces/1-image.jpg"}, "support": {"user": "George Bluth", "email": "george.bluth@reqres.in"}]}
 - Step 3: Attempts to update the user's name. Status 400. Response body: {"error": "Empty request body", "message": "Request body cannot be empty for JSON endpoints"}
 - Final message: Failed to update user.



The screenshot shows a test report titled "PlaywrightApiReport.html" with the following details:

- Tests:** A sidebar listing three test steps:
 - Step 1: Create a user and validate response** (Status: Passed, timestamp: 2:38:09 PM / 00:00:01:306)
 - Step 2: Get the created user details and validate** (Status: Passed, timestamp: 2:38:10 PM / 00:00:00:161)
 - Step 3: Update user's name and validate** (Status: Failed, timestamp: 2:38:10 PM / 00:00:00:508)
- Step 3: Update user's name and validate** (Details):

STATUS	TIMESTAMP	DETAILS
Info	2:38:11 PM	Update User - HTTP Status Code: 400
Fail	2:38:11 PM	Failed to update user.