STE Assessment — Professional Report

Contenido

1. Data

- **Login page:** https://wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/Account/Login
- **API base:** https://wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/api/employees
- **Valid credentials (test account):**
    - **Username:** TestUser802
    - **Password:** cf@ATyv+XI*{
    - **Token:** VGVzdFVzZXI4MDI6Y2ZAQVR5ditYSSp7

2. Login page — Bugs

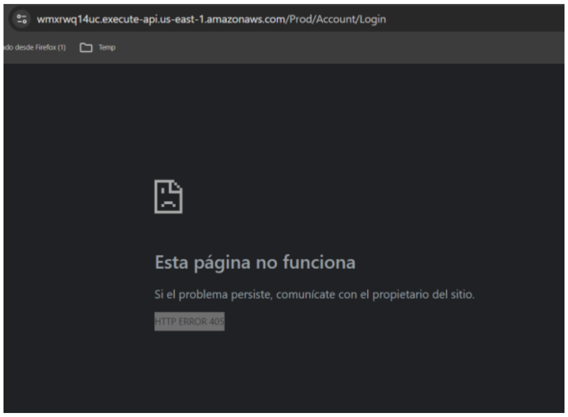
Bug 1 — No custom error page

- **Steps to reproduce**

    1. Enter an invalid username and password on the login page.

- **Expected result**
  A custom error page (e.g., "Invalid username or password").

- **Actual result**
  A generic error page is shown with HTTP ERROR 405.


wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/Account/Login
ado desde Firefox (1)
Temp

Esta página no funciona

Si el problema persiste, comunícate con el propietario del sitio.

HTTP ERROR 405

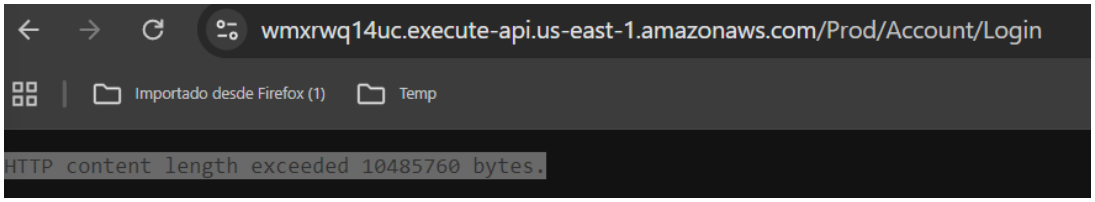
Bug 2 — Username field accepts extremely long input (2147483647)

- **Steps to reproduce**

    1. Enter a very long string into the username field (e.g., a string up to or beyond 2,147,483,647 characters).

- **Expected result**
  The username input should enforce a reasonable maximum length (for example 50–100 characters) and display a validation error if exceeded.

- **Actual result**
  The username field allows very large input and may lead to HTTP content length exceeded errors.


wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/Account/Login
Importado desde Firefox (1)
Temp

```
HTTP content length exceeded 10485760 bytes.
```

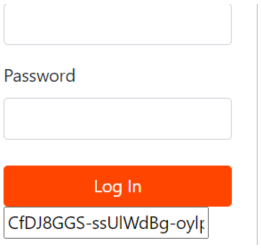
Bug 3 — Hidden verification token exposed in DOM

- **Steps to reproduce**

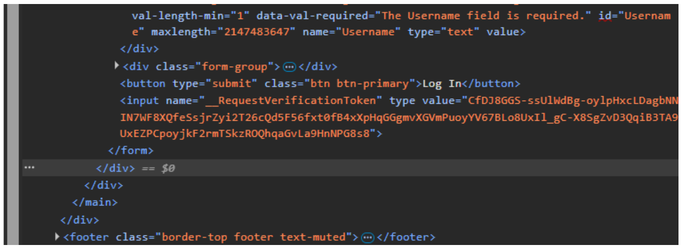
    1. Open browser developer tools.
    2. Search for "__RequestVerificationToken".
    3. Remove type="hidden" or otherwise manipulate the DOM.

- **Expected result**
  Sensitive tokens should not be exposed in the page where they can be easily manipulated

- **Actual result**
  The token is visible in the DOM.



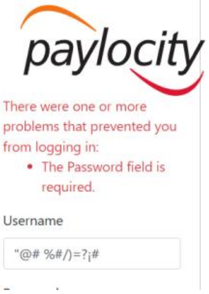
CfDJ8GGS-ssUlWdBg-oylp



```
        val-length-min="1" data-val-required="The Username field is required." id="Usernam
        e" maxlength="2147483647" name="Username" type="text" value>
       </div>
      ▸<div class="form-group">⋯</div>
       <button type="submit" class="btn btn-primary">Log In</button>
       <input name="__RequestVerificationToken" type value="CfDJ8GGS-ssUlWdBg-oylpHxcLDagbNN
       IN7WF8XQfeSsjrZyi2T26cQd5F56fxt0fB4xXpHqGGgmvXGVmPuoyYV67BLo8UxIl_gC-X8SgZvD3QqiB3TA9
       UxEZPCpoyjkF2rmTSkzROQhqaGvLa9HnNPG8s8">
      </form>
…    </div> == $0
    </div>
   </main>
  </div>
 ▸<footer class="border-top footer text-muted">⋯</footer>
```

Risk — Username allows arbitrary special characters

- **Recommendation**
  Restrict allowed characters for usernames to a safe subset (e.g., alphanumeric, dots, underscores). Always validate and sanitize on the server side.



There were one or more problems that prevented you from logging in:

- The Password field is required.

Username

"@# %#/)=?¡#

---

3. Benefits Dashboard — Bugs

Bug 1 — Mandatory fields not indicated

- **Steps to reproduce**
    1. Log in with valid credentials.
    2. Click **Add Employee**.
    3. Click **Add** without filling fields.
- **Expected result**
  Required fields should be marked and validation messages should appear if they are empty.
- **Actual result**
  No field changes and no error messages are shown.

Bug 2 — First name and last name accept special characters; no error shown

- **Steps to reproduce**
    1. Log in.
    2. Click **Add Employee**.
    3. Enter !"#$%$ in First Name and !"!"!"!" in Last Name, dependents 30.
    4. Click **Add**.
- **Expected result**
  Input with disallowed special characters should be rejected and appropriate messages should be shown; employee should not be added.
- **Actual result**
  A new employee is added despite invalid names.

Bug 3 — First and Last name maximum length not displayed

- **Steps to reproduce**
  1. Log in.
  2. Add an employee using First Name and Last Name strings longer than 50 characters.
- **Expected result**
  Validation message: “The field FirstName must be a string with a maximum length of 50.” (and similarly for LastName).
- **Actual result**
  No error messages shown

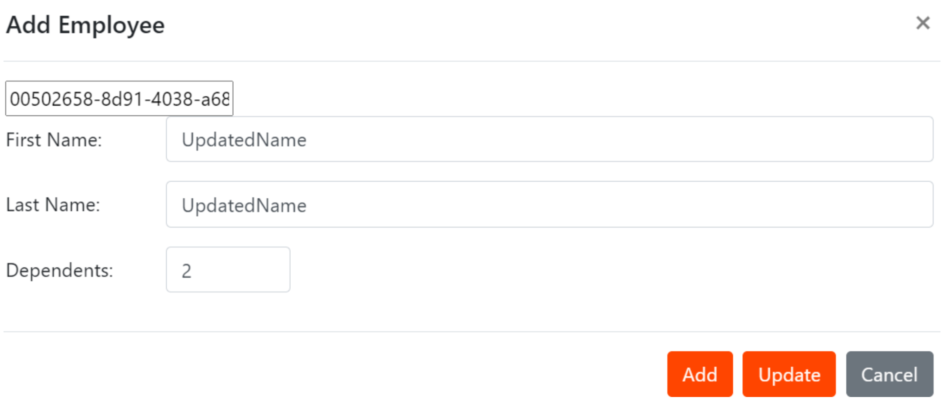
Bug 4 — Dependents field range not enforced

- **Steps to reproduce**
  1. Log in.
  2. Add an employee with dependents = 50.
- **Expected result**
  Error: “The field Dependents must be between 0 and 32.”
- **Actual result**
  No error messages

Bug 5 — Hidden id field and update control in DOM can be used to modify existing employees

- **Steps to reproduce**
  1. Log in.
  2. Click **Add Employee**.
  3. Inspect DOM, find id and updateEmployee inputs, remove type="hidden".
  4. Enter an existing employee id, modify other fields, click **Update**.
- **Expected result**
  The Add Employee form should not permit updating existing employees. Hidden fields should not be abused to perform updates. Server must validate intent (create vs update).

- **Actual result**
  Employee data is updated when manipulating hidden fields.



Add Employee

00502658-8d91-4038-a68

First Name: UpdatedName

Last Name: UpdatedName

Dependents: 2

Recommendation — Improve table (filtering, sorting, search)

- **Current issue**
  Employee listing relies primarily on ID and is hard to search or sort.
- **Recommendation**
  Add filters (by name, username,salary,etc), column sorting, free-text search, and pagination.

4. API Tests — Bugs

General schema concern

- **Observed**: username is treated as a required property in the Swagger but the API does not consistently enforce it.

GET /api/Employees/{id}

Error handling for invalid id format

- **Request example**
  GET /api/Employees/test (where test is not a UUID)
- **Expected result**
  API should validate the id format and return 400 Bad Request with a clear message or 404
- **Actual result**
  API returns 500 Internal Server Error when id is not a UUID.

POST /api/Employees

Missing required fields and invalid character acceptance

- **Request example (missing username)**

```
{
  "firstName": "first",
  "lastName": "second"
}
```

- **Expected result**
  Return 400 Bad Request stating username is required.
- **Actual result**
  API returns HTTP 200 OK and creates a record without username.

invalid character acceptance

Request example (special characters)

```
{
  "firstName": "##$^*()_+_)(*&^%$#@!",
  "lastName": "!@#$%^&*()_)(*&^%$#@!"
}
```

- **Expected result**
  Reject input containing invalid characters for name fields; return 400 with validation errors.

- **Actual result**
  API returns 200 OK and accepts the values.

POST /api/Employees —Incorrect HTTP Code

Request example (special characters)

```
{
  "firstName": "test",
  "lastName": "test",
   “username”: “TestUser802”
}
```

- **Expected result**
  Actual result
  API returns 201 Created and accepts the values.

- **Actual result**
  API returns 200 OK and accepts the values.

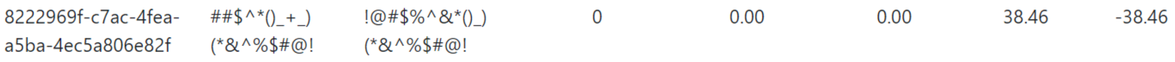
PUT /api/Employees/{id}

Incorrect create-on-put behavior and salary updates

- **Observed flaws**
  - PUT appears to create new employees if the id does not exist. Expected behavior for PUT to a specific resource is to update an existing resource or return 404 Not Found if it does not exist. Creation via PUT is allowed in some APIs but must be explicit and documented; current behavior is unexpected and problematic.
  - Salary and compensation fields can be updated by clients
- **Request example (PUT creating new employee)**

```
{
  "firstName": "first",
  "lastName": "last",
  "id": "8d9bff64-00e9-43d6-a94d-57183bca2290"
}
```

- **Expected result**
  If id does not exist, return 404 Not Found
- **Actual result**
  A new employee is added with salary and other numeric fields defaulting to zero.

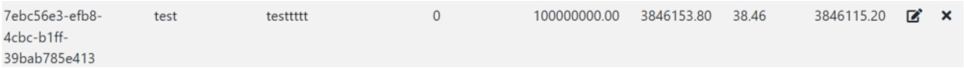


8222969f-c7ac-4fea-a5ba-4ec5a806e82f	##$^*()_+_) (*&^%$#@!	!@#$%^&*()_) (*&^%$#@!	0	0.00	0.00	38.46	-38.46

- **Request example (salary update)**

```
{
  "partitionKey": "TestUser802",
  "username": "TestUser802",
  "firstName": "test",
  "lastName": "testtttt",
  "salary": 100000000,
  "benefitsCost": 38.46154,
  "net": 1961.5385,
  "id": "7ebc56e3-efb8-4cbc-b1ff-39bab785e413"
}
```

- **Expected result**
  Salary and compensation fields should only be updatable by back-end processes or authorized clients
- **Actual result**
  Salary was updated by the API.



7ebc56e3-efb8-4cbc-b1ff-39bab785e413	test	testtttt	0	100000000.00	3846153.80	38.46	3846115.20	✎	×

- **Recommendation**
  - Enforce whether PUT creates vs updates by design; prefer PUT /api/Employees/{id} to update and return 404 if missing.
  - Return appropriate status codes: 200 OK for successful updates, 201 Created if creation is intentional, 400 for bad input, 404 when resource not found, 403 for unauthorized attempts.