ReadOnly PUT/POST error

ReadOnly values can be sent in PUT and POST requests

QA Eng: Diego Lizarraga
Testing Tool: Postman
Severity: Medium - High
Priority: (Usually PO handles this) P1 / P5
Environment: PROD
Version: 1
Endpoint: https://wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/api/Employees
Date: Friday, September 26, 2025

Description -1 (POST):

In the documentation, 5 properties should be readOnly, meaning that when the user tries to send them as part of the Request, there should be an error. If the request adds these 5 readOnly properties and the other required fields, the Employee is created, and the response is a 200 OK status.

Steps to Reproduce:

Pre-conditions: Access Postman, create the Post request with the endpoint URL, and add the Authorization header with the auth_token provided.

1. Add the Key: Content-Type, Value: application/json header
2. Add a raw JSON-formatted body that includes the required fields, and all the read-only fields.
3. Click on Send

Request Data:

```
{
   "partitionKey": "SomePartitionKey",
   "sortKey": "241d123-1456-4789-9012-b972cbd0e345",
   "username": "GREENLANTERN",
   "firstName": "GUY",
   "lastName": "GARDNER",
   "dependants": 0,
   "salary": 70000,
   "gross": 2000,
   "benefitsCost": 2000,
   "net": 4000
}
```

Expected Results:

The New Employee shouldn't be created, or the Response should be a 400 Bad Request or 405 Method Not Allowed, since some ReadOnly values were given a value.

Actual Results:

The New employee is created; however, all fields except firstName and lastName are different. And the system returns a 200 OK response.

```
{
    "partitionKey": "TestUser813",
    "sortKey": "b6397359-828a-446c-ae77-c9c2561f745f",
    "username": "TestUser813",
    "id": "b6397359-828a-446c-ae77-c9c2561f745f",
    "firstName": "GUY",
    "lastName": "GARDNER",
    "dependants": 0,
    "salary": 52000,
    "gross": 2000,
    "benefitsCost": 38.46154,
    "net": 1961.5385
}
```

Description - 2 (PUT):
In the documentation, 5 properties should be readOnly, meaning that when the user tries to send them as part of the Request, there should be an error. If the request adds these 5 readOnly properties and the other required fields, the Employee is created, and the response is a 200 OK status.

Steps to Reproduce:
Preconditions: Access Postman, create a PUT request with the endpoint URL, and add the Authorization header with the provided auth_token.

1. Add the Key: Content-Type, Value: application/json header
2. Add a raw JSON-formatted body that includes the required fields and all the read-only fields.
3. Click on Send

Request Data:

```
{
    "partitionKey": "SomePartitionKey",
    "sortKey": "241d123-1456-4789-9012-b972cbd0e345",
    "id": "b6397359-828a-446c-ae77-c9c2561f745f",
    "username": "GREENLANTERN",
    "firstName": "GUY",
    "lastName": "GARDNER",
    "dependants": 0,
    "salary": 70000,
    "gross": 2000,
    "benefitsCost": 2000,
    "net": 4000
}
```

Expected Results:
The New Employee shouldn't be updated, or the Response should be a 400 Bad Request or 405 Method Not Allowed, since some ReadOnly values were given a value.

Actual Results:
The New employee is partially updated (only the salary). And the system returns a 200 OK response.

```
{
    "partitionKey": "TestUser813",
    "sortKey": "b6397359-828a-446c-ae77-c9c2561f745f",
    "username": "TestUser813",
    "id": "b6397359-828a-446c-ae77-c9c2561f745f",
    "firstName": "GUY",
    "lastName": "GARDNER",
    "dependants": 0,
    "salary": 70000,
    "gross": 2692.3076,
    "benefitsCost": 38.46154,
    "net": 2653.8462
}
```

Username not required

Employee can be created without Username

QA Eng: Diego Lizarraga
Testing Tool: Postman
Severity: Medium
Priority: (Usually PO handles this) P3
Environment: PROD
Version: 1
Endpoint: https://wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/api/Employees
Date: Friday, September 26, 2025

Description (POST):
In the documentation, it is specified that the Username property is required, but the user can create a new register without a Username.
Steps to Reproduce:
Preconditions: Access Postman, create the Post request with the endpoint URL, and add the Authorization header with the auth_token provided.

1. Add the Key: Content-Type, Value: application/json header
2. Add a raw JSON-formatted body with only a firstName and lastName
3. Click on Send

Request Data:

```
{
   "firstName": "Noah",
   "lastName": "Wallberg"
}
```

Expected Results:
The New Employee shouldn't be created, or the Response should be a 400 Bad Request or 405 Method Not Allowed, since the username is required.

Actual Results:
The New employee is created with the username from the Authentication token. And the system returns a 200 OK response.

```
{
   "partitionKey": "TestUser813",
   "sortKey": "c00e4ab8-7dec-4317-863d-0f8d13be5b11",
   "username": "TestUser813",
   "id": "c00e4ab8-7dec-4317-863d-0f8d13be5b11",
   "firstName": "Noah",
   "lastName": "Wallberg",
   "dependants": 0,
   "salary": 52000,
   "gross": 2000,
   "benefitsCost": 38.46154,
   "net": 1961.5385
}
```

Overwritten Username

Username overwritten from the Authentication token

QA Eng: Diego Lizarraga
Testing Tool: Postman
Severity: Medium
Priority: (Usually PO handles this) P3
Environment: PROD
Version: 1
Endpoint: https://wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/api/Employees
Date: Friday, September 26, 2025

Description -1 (POST):
According to the documentation, the Username property is required, meaning each Employee should have at least the opportunity to use a different username.

Steps to Reproduce:
Pre-conditions: Access Postman, create the Post request with the endpoint URL, and add the Authorization header with the auth_token provided.

1. Add the Key: Content-Type, Value: application/json header
2. Add a raw JSON-formatted body with only firstName, lastName, and username
3. Click on Send

Request Data:

```
{
   "firstName": "Chilly",
   "lastName": "Willy",
   “username”: “VeryChill”
}
```

Expected Results:
The New Employee is created with the Request data fields.

Actual Results:
The New employee is created with the username from the Authentication token.

```
{
   "partitionKey": "TestUser813",
   "sortKey": "6375e220-9bfd-4e21-a9d9-d791814608e2",
   "username": "TestUser813",
   "id": "6375e220-9bfd-4e21-a9d9-d791814608e2",
   "firstName": "Chilly",
   "lastName": "Willy",
   "dependants": 0,
   "salary": 52000,
   "gross": 2000,
   "benefitsCost": 38.46154,
   "net": 1961.5385
}
```

Description -2 (PUT):
According to the documentation, the Username property is required, meaning each Employee should have a Unique Username.

Steps to Reproduce:

Pre-conditions: Access Postman, create the PUT request with the endpoint URL, and add the Authorization header with the auth_token provided.

1. Add the Key: Content-Type, Value: application/json header
2. Add a raw JSON-formatted body with only firstName, lastName, and username
3. Click on Send

Request Data:

```
{
    "firstName": "Not Chilly",
    "lastName": "Willy",
    “username”: “NOTCHILLATALL”,
    "id": "6375e220-9bfd-4e21-a9d9-d791814608e2"
}
```

Expected Results:

The New Employee is updated correctly, and the changes are shown accordingly

Actual Results:

The New employee is edited, but the username keeps the value from the Authentication token.

```
{
    "partitionKey": "TestUser813",
    "sortKey": "6375e220-9bfd-4e21-a9d9-d791814608e2",
    "username": "TestUser813",
    "id": "6375e220-9bfd-4e21-a9d9-d791814608e2",
    "firstName": "Not Chilly",
    "lastName": "Willy",
    "dependants": 0,
    "salary": 52000,
    "gross": 2000,
    "benefitsCost": 38.46154,
    "net": 1961.5385
}
```

Salary should not be updated

Salary can be updated to invalid values

QA Eng: Diego Lizarraga
Testing Tool: Postman
Severity: High
Priority: (Usually PO handles this) P1
Environment: PROD
Version: 1
Endpoint: https://wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/api/Employees
Date: Friday, September 26, 2025

Description (PUT):

In the assumptions, it is specified that all employees are paid 2000 per paycheck, and that there are 26 paychecks in a year. Meaning that the salary should always be 52000. When the Employer edits an Employee, the salary can range from any value, including negative numbers and large positive numbers.

Steps to Reproduce:

Preconditions: Access Postman, create a PUT request with the endpoint URL, and add the Authorization header with the provided auth_token. Create a new Employee.

1. Add the Key: Content-Type, Value: application/json header
2. Add a raw JSON-formatted body with firstName, lastName, and id of the previously created employee
3. Add an invalid number as salary into the request data
4. Click on Send

Request Data:

```
{
   "firstName": "Not Chilly",
   "lastName": "Willy",
   "id": "6375e220-9bfd-4e21-a9d9-d791814608e2",
   "salary": -100000}
```

Expected Results:

There should be a control where the salary should not be modified, or as a functioning readOnly property, thus resulting in a 400 Bad Request or 405 Method Not Allowed status.

Actual Results:

The salary gets modified to an invalid value, and the system returns a 200 OK status

```
   "id": "6375e220-9bfd-4e21-a9d9-d791814608e2",
   "firstName": "Not Chilly",
   "lastName": "Willy",
   "dependants": 0,
   "salary": -100000,
   "gross": -3846.1538,
   "benefitsCost": 38.46154,
   "net": -3884.6152
}
```

Deleting non-existent employee

Deleting a non-existent Employee returns a 200 OK status

QA Eng: Diego Lizarraga
Testing Tool: Postman
Severity: Low
Priority: (Usually PO handles this) P5
Environment: PROD
Version: 1
Endpoint:
https://wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/api/Employees{id}
Date: Friday, September 26, 2025

Description (PUT):
When a user tries to delete a previously deleted or non-existent Employee, the system returns a 200 OK status.

Steps to Reproduce:
Pre-conditions: Access Postman, create the DELETE request with the endpoint URL, and add the Authorization header with the auth_token provided. Create a new Employee.

1. Add the Key: Content-Type, Value: application/json header
2. Add the ID of the previously created Employee at the end of the endpoint
3. Click on Send
4. Validate that the Employee is no longer registered
5. Click on Send again

Expected Results:
The system should return a 404 Not Found status, ideally with a "Employee not found" message.

Actual Results:
The system returns a 200 OK status.

GET non-existent employee

Retrieving a non-existent Employee returns a 200 OK status

QA Eng: Diego Lizarraga
Testing Tool: Postman
Severity: Low
Priority: (Usually PO handles this) P5
Environment: PROD
Version: 1
Endpoint:
https://wmxrwq14uc.execute-api.us-east-1.amazonaws.com/Prod/api/Employees/{id}
Date: Friday, September 26, 2025

Description (PUT):
When a user tries to GET a previously deleted or non-existent Employee, the system returns a 200 OK status.

Steps to Reproduce:
Pre-conditions: Access Postman, create the GET request with the endpoint URL, and add the Authorization header with the auth_token provided. Delete an Employee.

1. Add the Key: Content-Type, Value: application/json header
2. Add the ID of the previously deleted Employee at the end of the endpoint
3. Click on Send

Expected Results:
The system should return a 404 Not Found status, ideally with a "Employee not found" message.

Actual Results:
The system returns a 200 OK status.