API Bugs:

Bug1: Incorrect Status Code on Duplicate Deletion

Severity: Medium

Description: The DELETE endpoint returns a `200 OK` status even when attempting to delete a resource that has already been removed.

Steps to Reproduce:

- Create a new employee via POST to obtain a valid `id`. Post Payload example

```
{
  "firstName": "test1",
  "lastName": "nayak1",
  "username": "tj1",
   "dependants": 4
 }
```

Response Body

```
{
    "partitionKey": "TestUser893",
    "sortKey": "caa0c439-f1f3-4df4-bb36-895a94ef59e3",
    "username": "TestUser893",
    "id": "caa0c439-f1f3-4df4-bb36-895a94ef59e3",
    "firstName": "test1",
    "lastName": "nayak1",
    "dependants": 4,
    "salary": 52000,
    "gross": 2000,
    "benefitsCost": 115.38462,
    "net": 1884.6154
}
```

- Execute a DELETE request for that `id` `caa0c439-f1f3-4df4-bb36-895a94ef59e3`
- Observe a `200 OK` response.
- Execute the same DELETE request for the same `id` a second time.

Actual Result: The system returns `200 OK` for the second deletion attempt.
Expected Result: The system should return `404 Not Found` for subsequent deletion attempts as the resource no longer exists.

Bug 2: Non-Standard Status Code for Successful Deletion

Severity: Low

Description: The DELETE endpoint returns `200 OK` upon successful removal of a record instead of the industry-standard status code for successful deletions with no returned body.

Steps to Reproduce:

- Create a new employee via POST to obtain a valid `id`. Post Payload example
- Execute a DELETE request for that `id` `caa0c439-f1f3-4df4-bb36-895a94ef59e3`

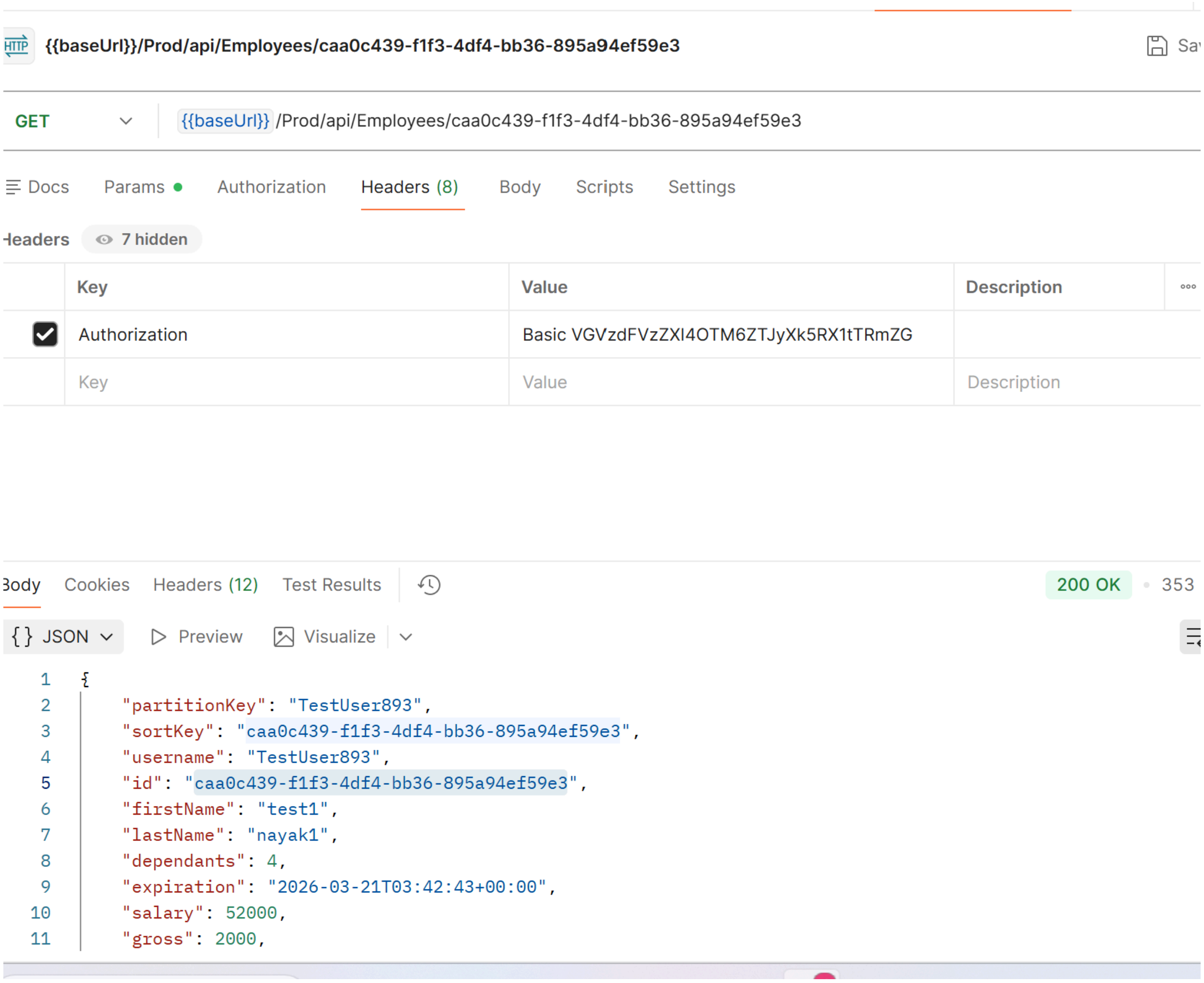
- Observe a `200 OK` response.

Actual Result: Returns `200 OK`.
Expected Result: Should return `204 No Content` to indicate successful processing with no response body

Bug 3: Stale Data Retrieval – GET Request Returns Success for Deleted Resources

Severity: Medium **Description:** The GET (by ID) endpoint returns a success status for resources that have been previously deleted from the dashboard or API.
Steps to Reproduce:

- Perform a POST call to create an employee and retrieve the ID.
- Perform a GET request by ID to verify the user exists.



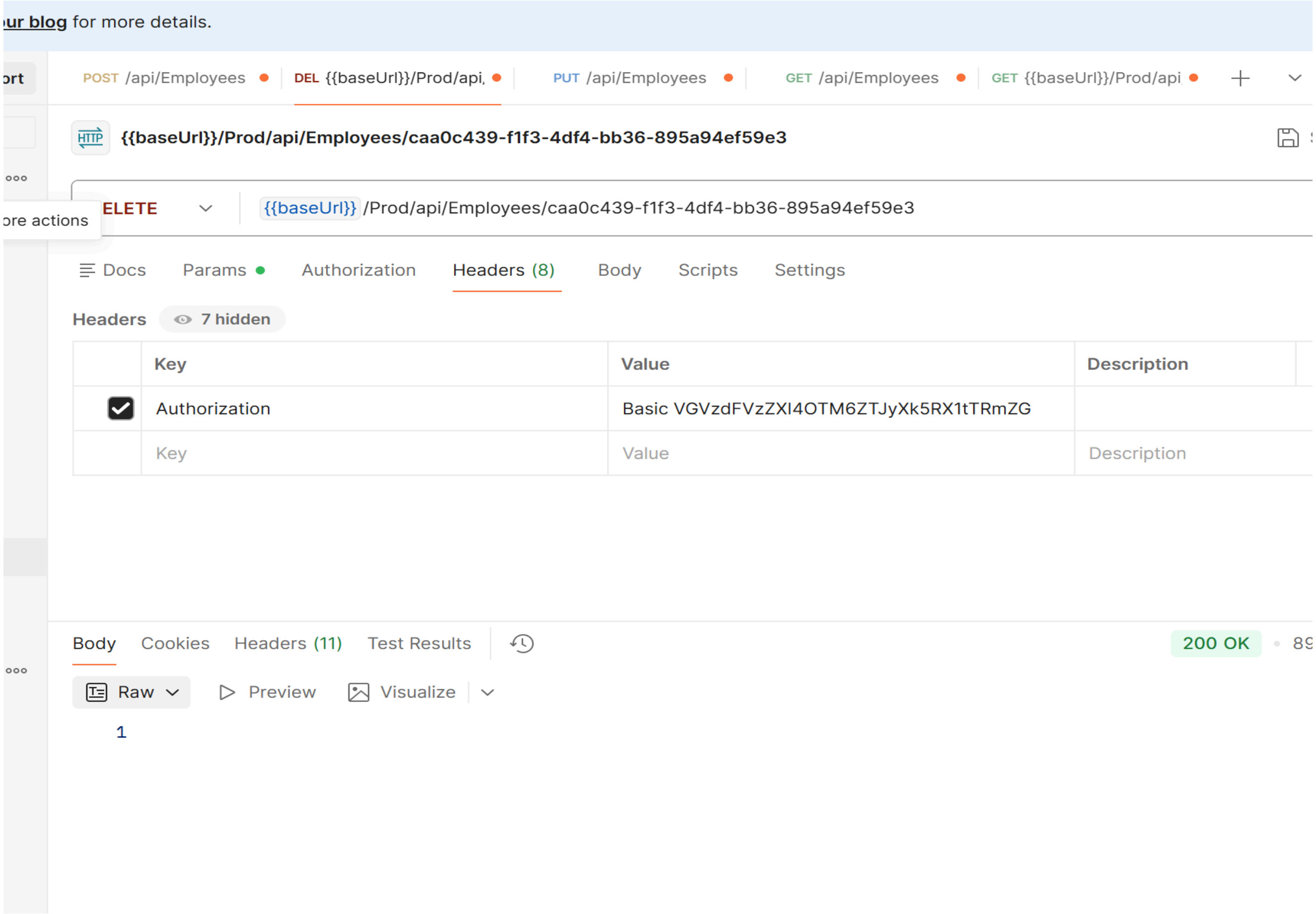
{{baseUrl}}/Prod/api/Employees/caa0c439-f1f3-4df4-bb36-895a94ef59e3

GET {{baseUrl}} /Prod/api/Employees/caa0c439-f1f3-4df4-bb36-895a94ef59e3

	Key	Value	Description
☑	Authorization	Basic VGVzdFVzZXI4OTM6ZTJyXk5RX1tTRmZG	
	Key	Value	Description

```
{
    "partitionKey": "TestUser893",
    "sortKey": "caa0c439-f1f3-4df4-bb36-895a94ef59e3",
    "username": "TestUser893",
    "id": "caa0c439-f1f3-4df4-bb36-895a94ef59e3",
    "firstName": "test1",
    "lastName": "nayak1",
    "dependants": 4,
    "expiration": "2026-03-21T03:42:43+00:00",
    "salary": 52000,
    "gross": 2000,
```

- Perform a DELETE request for that ID.



ur blog for more details.

{{baseUrl}}/Prod/api/Employees/caa0c439-f1f3-4df4-bb36-895a94ef59e3

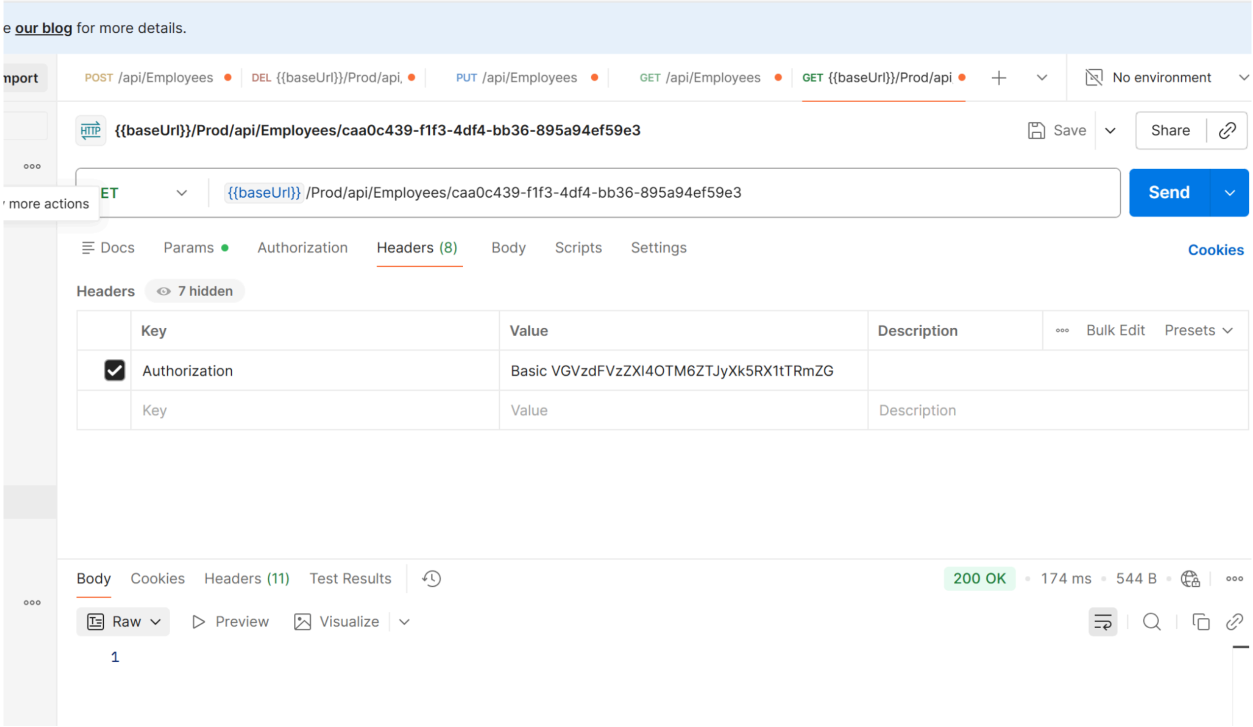
ELETE {{baseUrl}} /Prod/api/Employees/caa0c439-f1f3-4df4-bb36-895a94ef59e3

	Key	Value	Description
☑	Authorization	Basic VGVzdFVzZXI4OTM6ZTJyXk5RX1tTRmZG	
	Key	Value	Description

200 OK

```
1
```

- Repeat the GET request for the same ID.



e our blog for more details.

POST /api/Employees DEL {{baseUrl}}/Prod/api, PUT /api/Employees GET /api/Employees GET {{baseUrl}}/Prod/api

No environment

{{baseUrl}}/Prod/api/Employees/caa0c439-f1f3-4df4-bb36-895a94ef59e3

ET {{baseUrl}} /Prod/api/Employees/caa0c439-f1f3-4df4-bb36-895a94ef59e3

Headers (8)

Headers 7 hidden

	Key	Value	Description
☑	Authorization	Basic VGVzdFVzZXI4OTM6ZTJyXk5RX1tTRmZG	
	Key	Value	Description

Body Cookies Headers (11) Test Results 200 OK • 174 ms • 544 B

```
1
```

Actual Result: User receives a `200 OK` response with blank results.
Expected Result: User should receive a `404 Not Found` error when the employee has been successfully deleted.

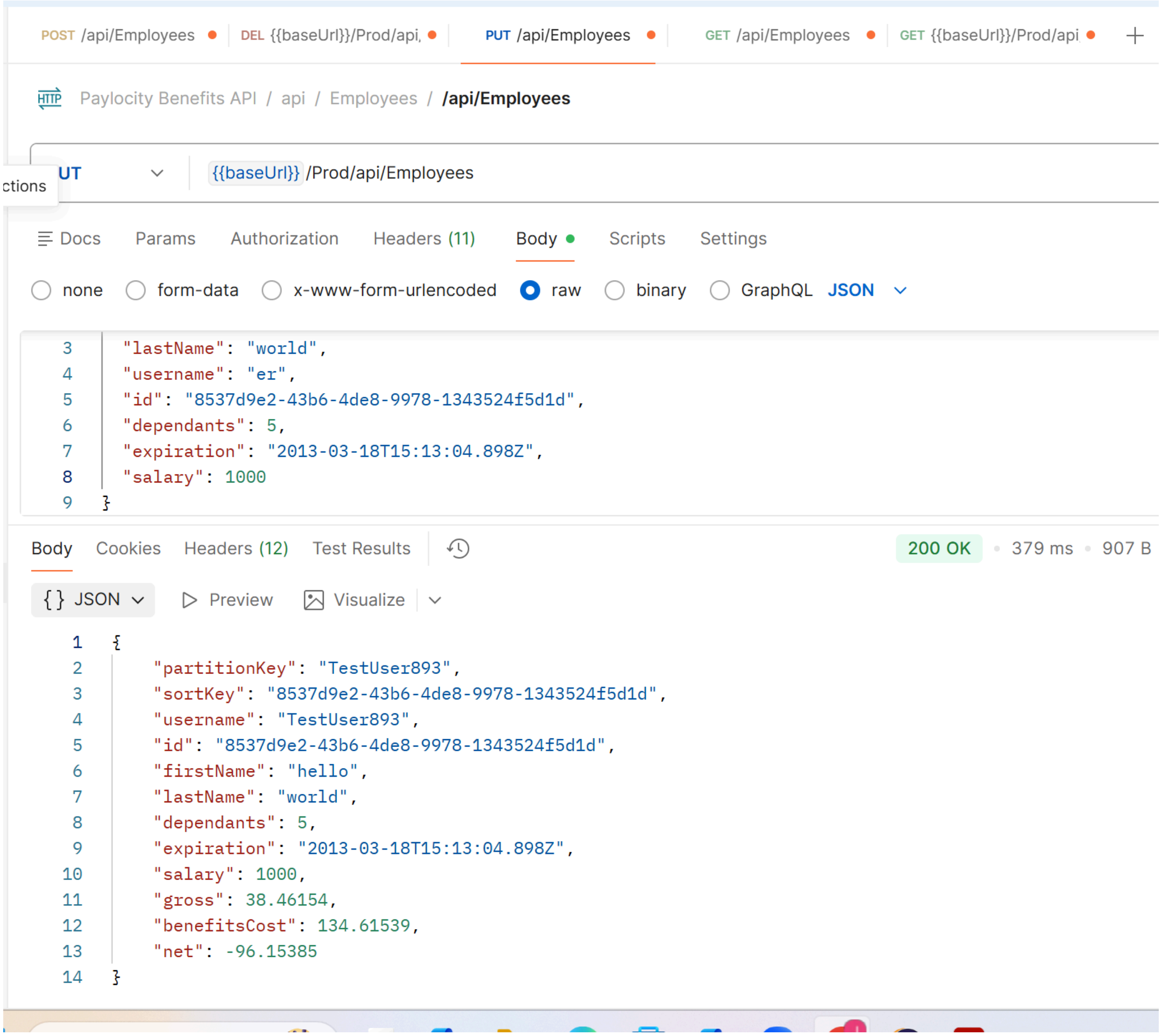
Bug 4: Missing Business Logic Validation – Negative Net Pay Calculation

Severity: High

Description: The API allows updating a record such that the total benefit costs exceed the employee's salary, resulting in a negative `net` pay value.
Steps to Reproduce

- Go to put call use a payload displayed below where salary is less than benefits or salary property is removed and hit send

Actual Result: The system calculates and persists a negative Net Pay (e.g., `-96.15`).
Expected Result: The API should include a validation check to prevent the salary from being lower than the annual benefit costs ($653.84 for maximum dependents and includes employee benefits), or enforce the requirement that salary remains defaulted to 52000.
Image attached below



Paylocity Benefits API / api / Employees / **/api/Employees**

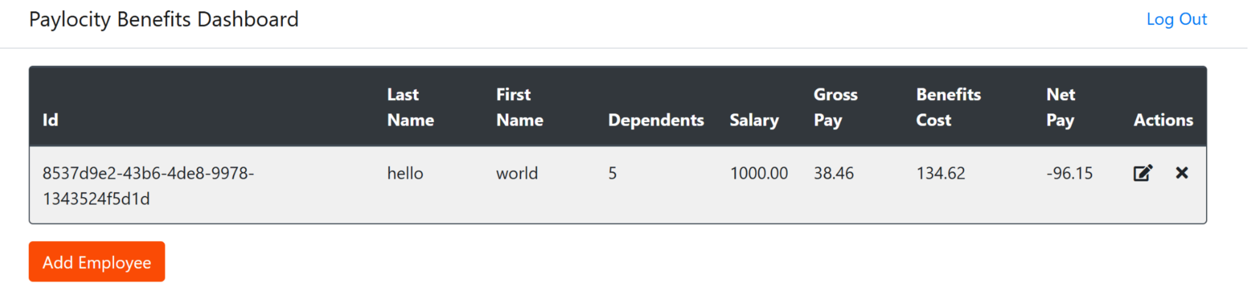
UT {{baseUrl}} /Prod/api/Employees

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
  "lastName": "world",
  "username": "er",
  "id": "8537d9e2-43b6-4de8-9978-1343524f5d1d",
  "dependants": 5,
  "expiration": "2013-03-18T15:13:04.898Z",
  "salary": 1000
}
```

200 OK • 379 ms • 907 B

```
{
    "partitionKey": "TestUser893",
    "sortKey": "8537d9e2-43b6-4de8-9978-1343524f5d1d",
    "username": "TestUser893",
    "id": "8537d9e2-43b6-4de8-9978-1343524f5d1d",
    "firstName": "hello",
    "lastName": "world",
    "dependants": 5,
    "expiration": "2013-03-18T15:13:04.898Z",
    "salary": 1000,
    "gross": 38.46154,
    "benefitsCost": 134.61539,
    "net": -96.15385
}
```



Paylocity Benefits Dashboard

Id	Last Name	First Name	Dependents	Salary	Gross Pay	Benefits Cost	Net Pay	Actions
8537d9e2-43b6-4de8-9978-1343524f5d1d	hello	world	5	1000.00	38.46	134.62	-96.15	✎ ✕

Bug 5: Incorrect Status Code on Successful Resource Creation

Severity: Low
Description: The POST endpoint returns a `200 OK` status when a new employee record is successfully created.
Steps To Reproduce

- Create a new employee via POST to obtain a valid `id`.
  Post Payload example

```
{
  "firstName": "test1",
  "lastName": "nayak1",
  "username": "tj1",
   "dependants": 4
 }
```

Actual Result: Returns `200 OK`.
Expected Result: Should return `201 Created` to correctly reflect the creation of a new resource.

Bug 6: Logic Inconsistency – PUT Request Performing Unexpected Create (Upsert)

Severity: Low

Description: When a PUT request is sent with a valid but non-existent GUID, the system creates a new record instead of rejecting the update.

Steps To Reproduce

- Send a valid payload but with random guid which is valid guid and not present in the dashboard

```
{
  "firstName": "hello2",
  "lastName": "worl2d",
  "username": "er",
  "id": "8587d9e2-43b6-4de8-9978-1343524f5d1d",
  "dependants": 5,
  "salary": 52000
}
```

Actual Result: The system adds a new employee and assigns a new ID.

Expected Result: Unless Upsert is an intended feature, the system should return `404 Not Found` because the targeted ID does not exist.
Note -> Bug can be ignored if upsert is expected

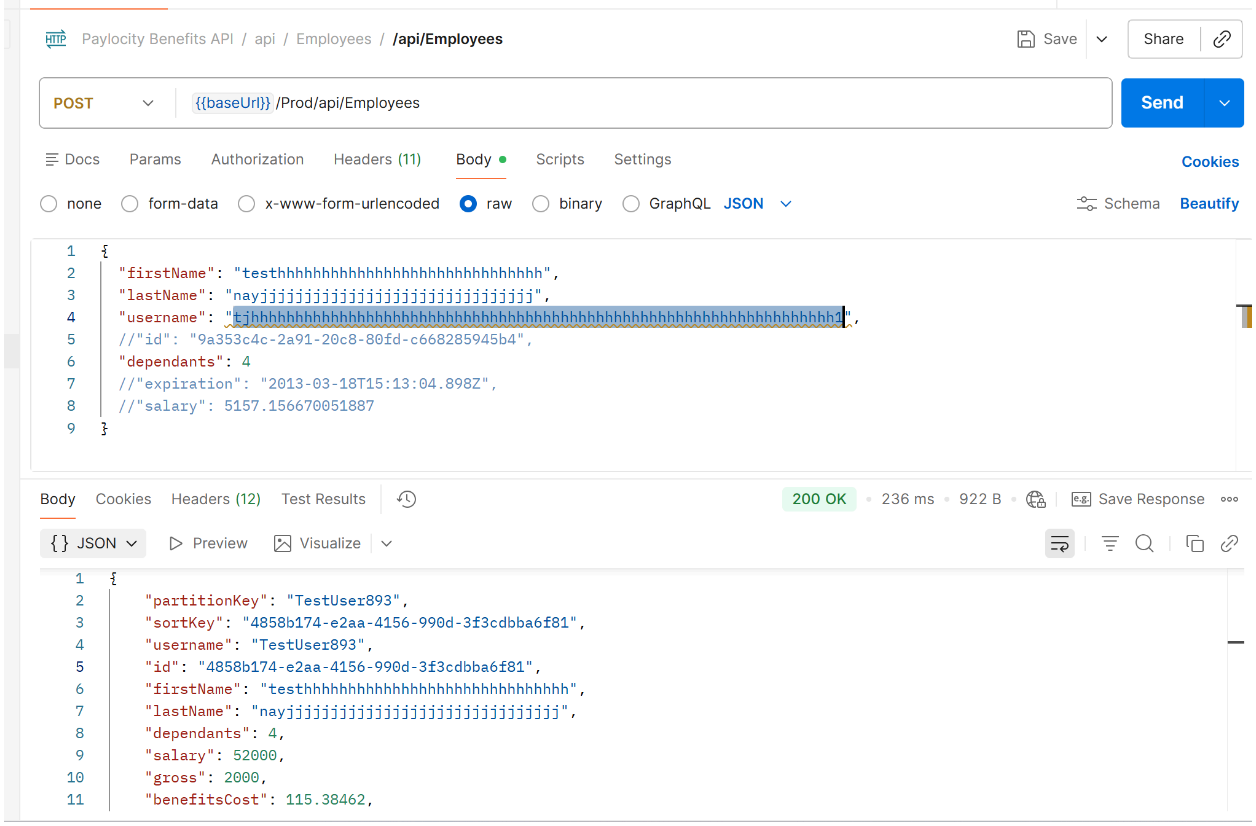
Bug 7: UserName property in the payload for put & postDoes not respect 50 character limit

Steps To Reproduce

- **Enter a valid post/put payload**

Post Payload example

```
{
  "firstName": "test1",
  "lastName": "nayak1",
  "username":
"tjhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh1
",
   "dependants": 4
 }
```



Paylocity Benefits API / api / Employees / /api/Employees

POST {{baseUrl}} /Prod/api/Employees

```
{
  "firstName": "testhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh",
  "lastName": "nayjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj",
  "username": "tjhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh1",
  //"id": "9a353c4c-2a91-20c8-80fd-c668285945b4",
  "dependants": 4
  //"expiration": "2013-03-18T15:13:04.898Z",
  //"salary": 5157.156670051887
}
```

200 OK • 236 ms • 922 B

```
{
    "partitionKey": "TestUser893",
    "sortKey": "4858b174-e2aa-4156-990d-3f3cdbba6f81",
    "username": "TestUser893",
    "id": "4858b174-e2aa-4156-990d-3f3cdbba6f81",
    "firstName": "testhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh",
    "lastName": "nayjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj",
    "dependants": 4,
    "salary": 52000,
    "gross": 2000,
    "benefitsCost": 115.38462,
```

Actual Result: The `username` field is currently required in the request body but is functionally useless as the system defaults to the logged-in user's credentials regardless of the input.

Expected Result: The `username` field should be removed from the request body schema to streamline the API and prevent potential data mismatch between the payload and the authenticated session.