

Task 5 :

BDD API test cases:

Test Case 1: Valid Request

- Scenario: Create a valid object with all required and optional fields within the specified limits.
- Given the API is accessible
- When a POST request is sent to " <https://restful-api.dev/> "

with the following payload:

```
[  
  name: "Unique Name,"  
  "data :[  
    "  year": 2023,  
    "  price": 999.99,  
    "  CPU model: "Intel Core i9,"  
    "  Hard disk size": "1 TB"  
  ]
```

- Then the response should have a status code of 201 (Created)

And the response body should contain the created object with the provided data

And the created object should have a unique ID

And the created object should match the provided payload

Test Case 2: Name Length Validation

- Scenario: Attempt to create an object with a name exceeding the maximum length limit.
- Given the API is accessible
- When a POST request is sent to "<https://restful-api.dev/>" with the following payload:

```
[  
  " name": "ThisIsAVeryLongNameThatExceedsTheMaxLengthLimit,"  
  "data :[  
    " year": 2022,  
    " price": 999.99,  
    " CPU model": "Intel Core i7,"  
    " Hard disk size": "512 GB"  
  ]
```

- Then the response should have a status code of 400 (Bad Request)

And the response body should contain an error message indicating the name length limit

Test Case 3: Name Uniqueness Validation

- Scenario: Attempt to create an object with a name that already exists.
- Given the API is accessible
- When a POST request is sent to " <https://restful-api.dev/> "with the following payload:

```
}  
" name": "Apple MacBook Pro 16,"  
"data :[  
"  year": 2022,  
"  price": 999.99,  
"  CPU model": "Intel Core i7,"  
"  Hard disk size": "512 GB"  
{
```

- Then the response should have a status code of 400 (Bad Request)

And the response body should contain an error message indicating the name should be unique

Test Case 4: Year Range Validation

- Scenario: Attempt to create an object with a year value outside the allowed range.
- Given the API is accessible
- When a POST request is sent to " <https://restful-api.dev/> "with the following payload:

```
}  
" name": "UniqueName,"  
"" data :[  
"  year": 1899,  
"  price": 999.99,  
"  CPU model": "Intel Core i7,"  
"  Hard disk size": "512 GB"  
{
```

- Then the response should have a status code of 400 (Bad Request)

And the response body should contain an error message indicating the year range limitation

Test Case 5: Price Validation

- Scenario: Attempt to create an object with an invalid price value.
- Given the API is accessible
- When a POST request is sent to " <https://restful-api.dev/> " with the following payload:

```
}  
" name": "UniqueName,"  
"data :[  
"  year": 2022,  
"  price": "InvalidPrice,"  
"  CPU model": "Intel Core i7,"  
"  Hard disk size": "512 GB"  
{
```

- Then the response should have a status code of 400 (Bad Request)

And the response body should contain an error message indicating the invalid price format

Test Case 6: Optional Fields

- Scenario: Create an object with only the required fields, without providing the optional fields.
- Given the API is accessible
- When a POST request is sent to " <https://restful-api.dev/> " with the following payload:

```
}
```

```
" name": "UniqueName,"
```

```
"data :[
```

```
"  year": 2022,
```

```
"  price": 999.99
```

```
{
```

- Then the response should have a status code of 201 (Created)

And the response body should contain the created object with the required fields

And the created object should have a unique ID

And the created object should match the provided payload, including the absence of the optional field.