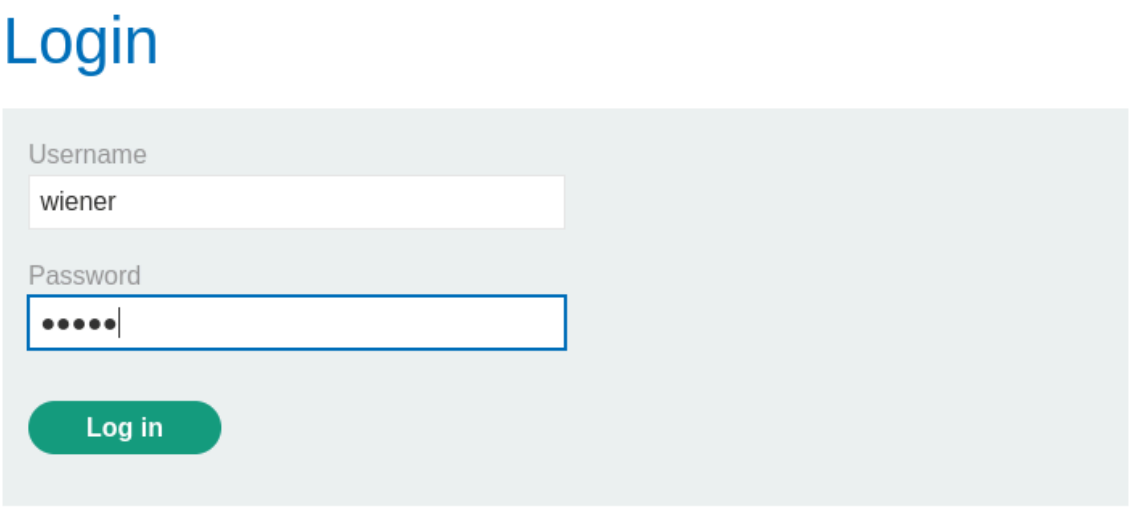
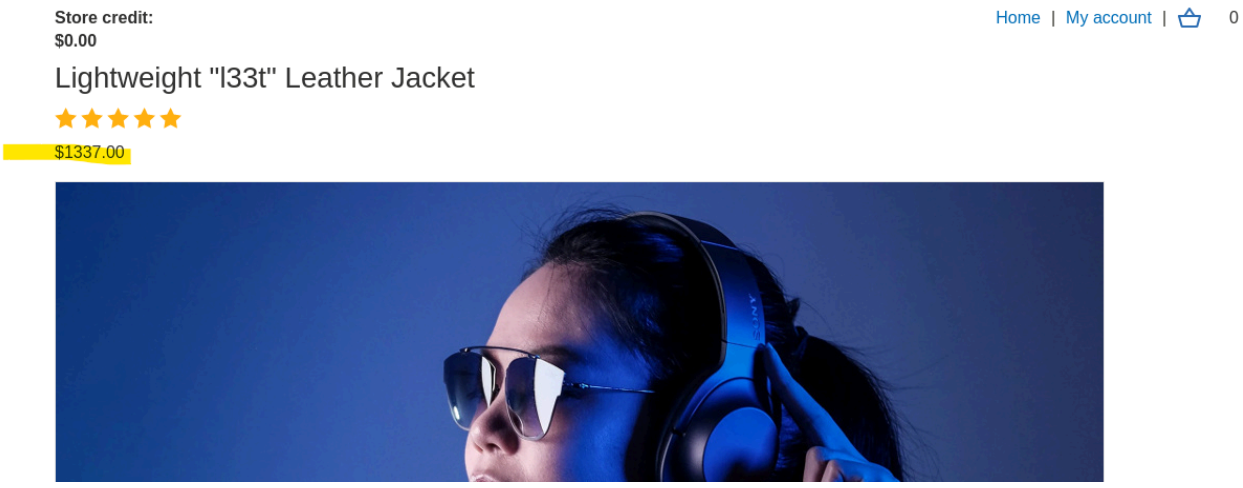
Lab link : https://portswigger.net/web-security/api-testing/lab-exploiting-unused-api-endpoint

---

1. **Enter the credentials**



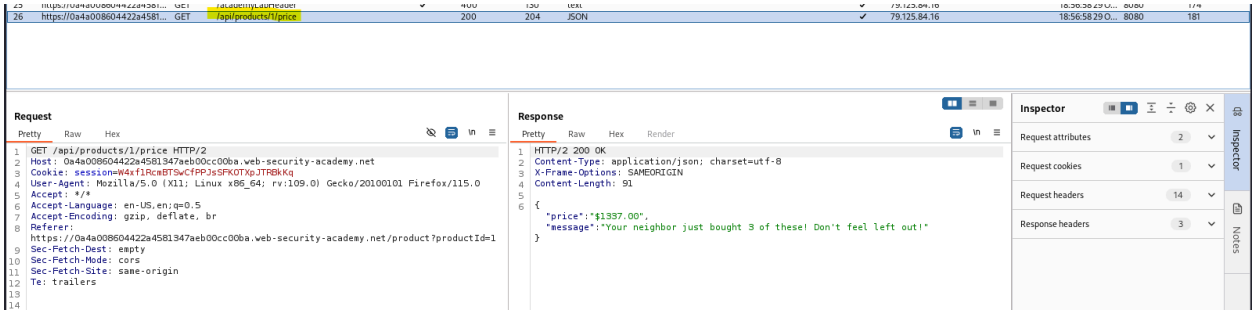
2. **Go to the leather jacket and see the price, and then add it to the cart. While doing this, make sure that Burp Suite is on.**



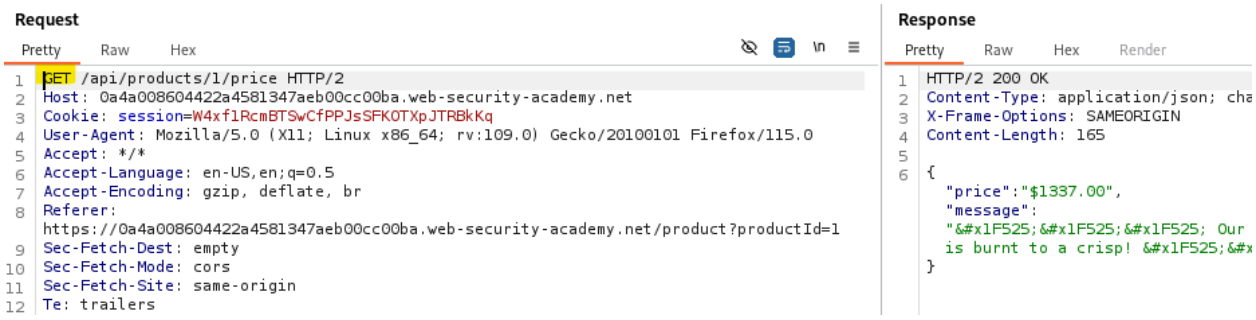
3. **Go to history in Burp Suite and locate the api.**

The api would look like this :

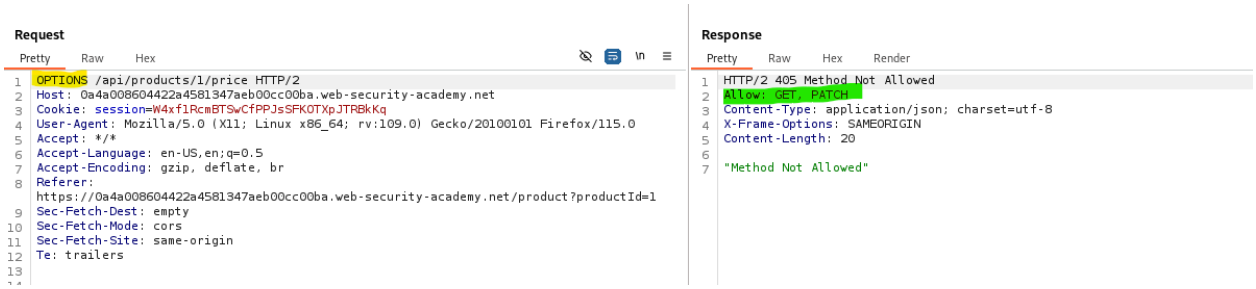
GET /api/products/price HTTP/2



4. **Next, send this to the repeater and change the value of GET to OPTIONS.**

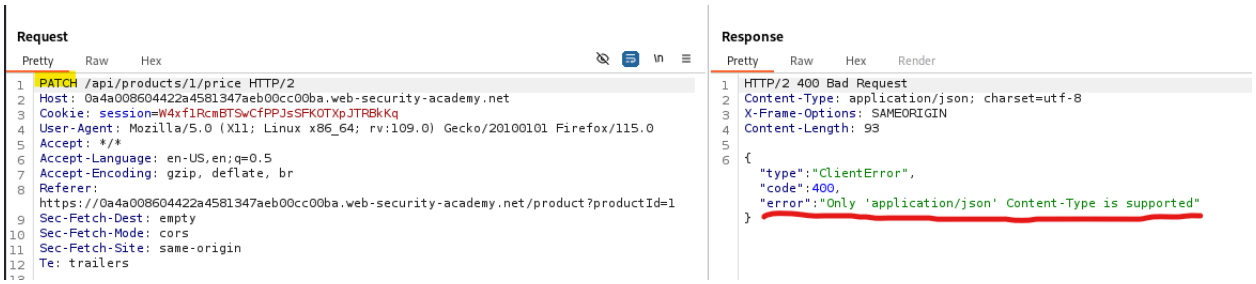


Then check the results using OPTIONS.



As indicated by the green highlighter above, the allowed methods are GET and PATCH. Therefore, I need to check the response when using the PATCH method.

5. **Change the OPTIONS to PATCH. To see what response it will give.**



Once it was changed from OPTIONS to PATCH. An error was shown that

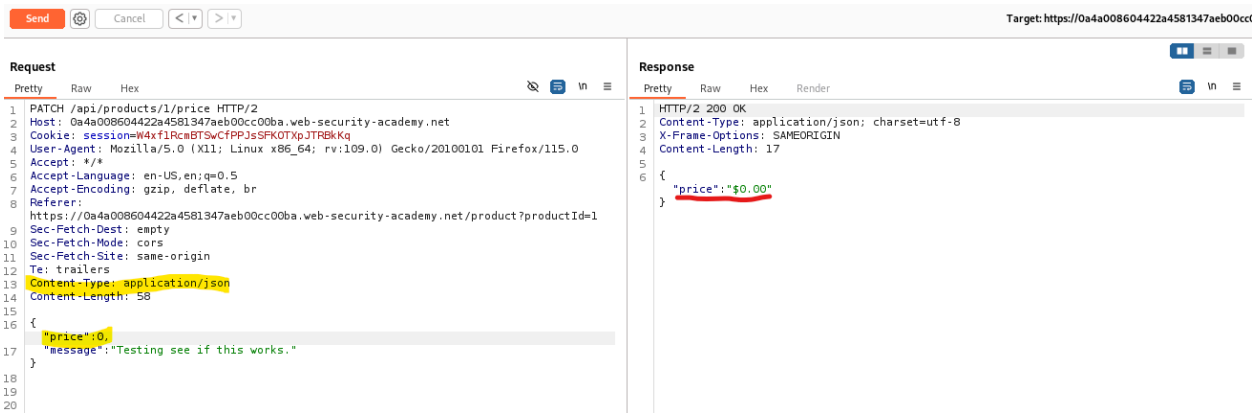
“Only application/json’ Content-type is supported”.

This means that the content header must be added.

6. **When adding the header for the content-type, make sure to include the request body as JSON. Setting the content-type to application/json indicates that the request body must be formatted as valid JSON.**

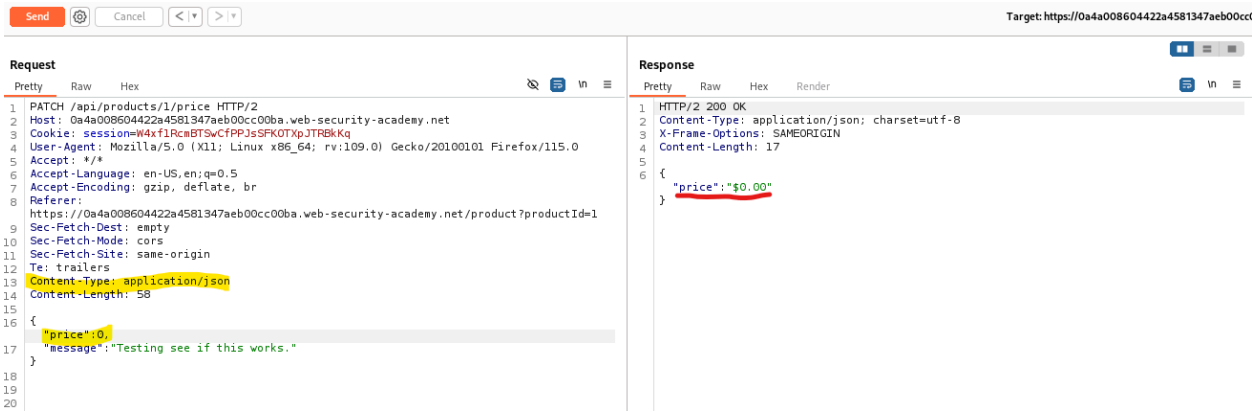
Add:

```
{
  "price": "0.00",
  "message": "testing see if it works"
}
```



As indicated by the yellow highlighter above, since the price parameter must be a non-negative number, change the price from $0.00 to just an integer 0.

7. **After changing the number, the response would be updated,**



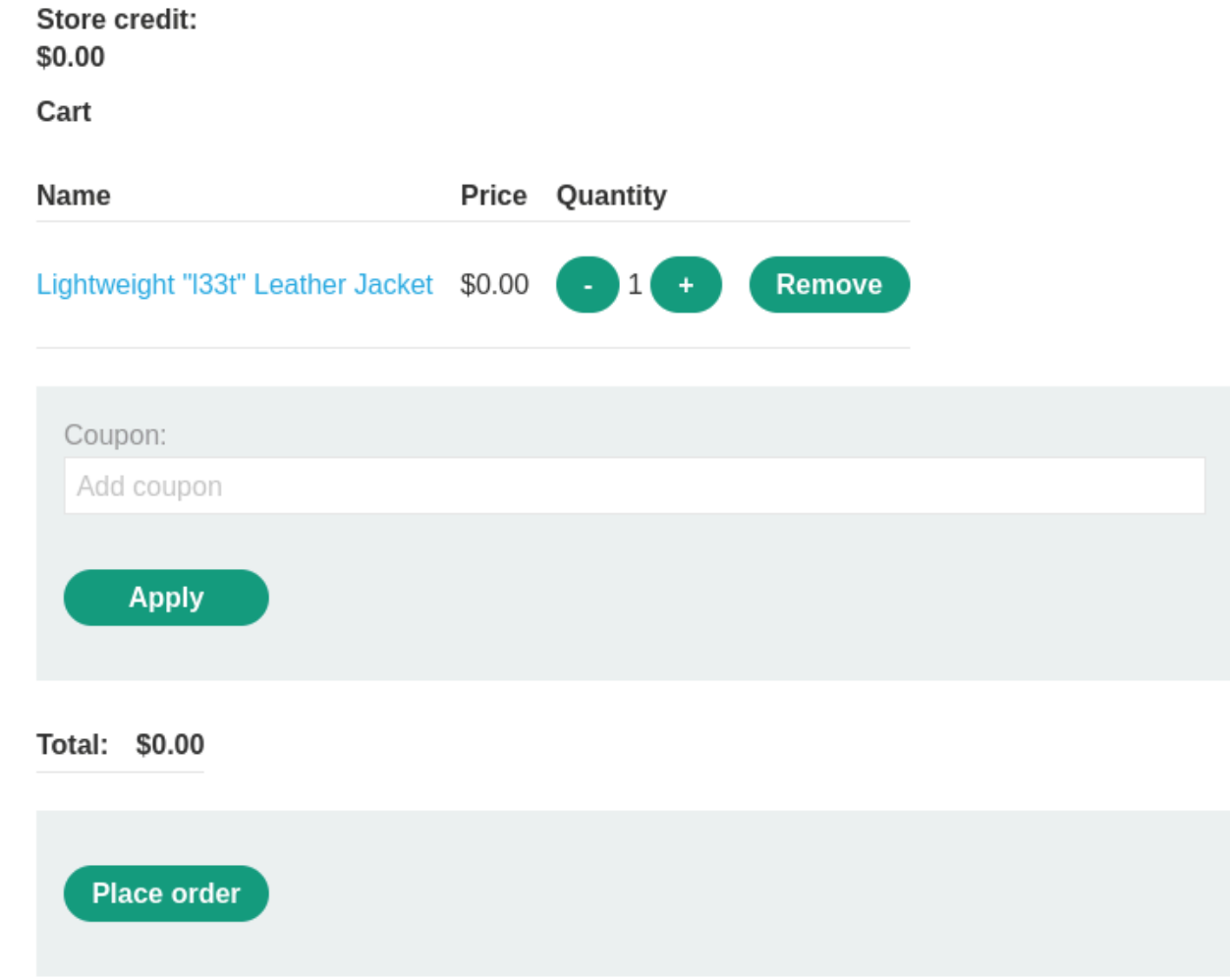
8. **Now go back to the home page and see the price of the leather jacket and add it to the cart.**

Lightweight "l33t" Leather Jacket

$0.00

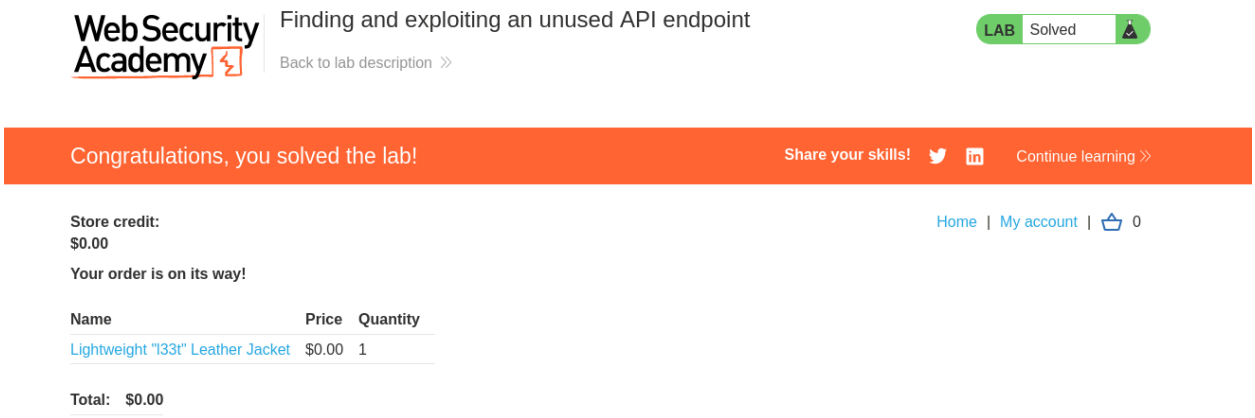
View details

This shows that the price was updated.



When added to the cart the price is $0.00.

Therefore ….



A free jacket its on its way and the lab is solved! 😉

💡What have I learned?

In API there are 8 HTTP methods.

- GET
- PUT
- POST
- DELETE
- PATCH
- HEAD
- OPTIONS
- TRACE

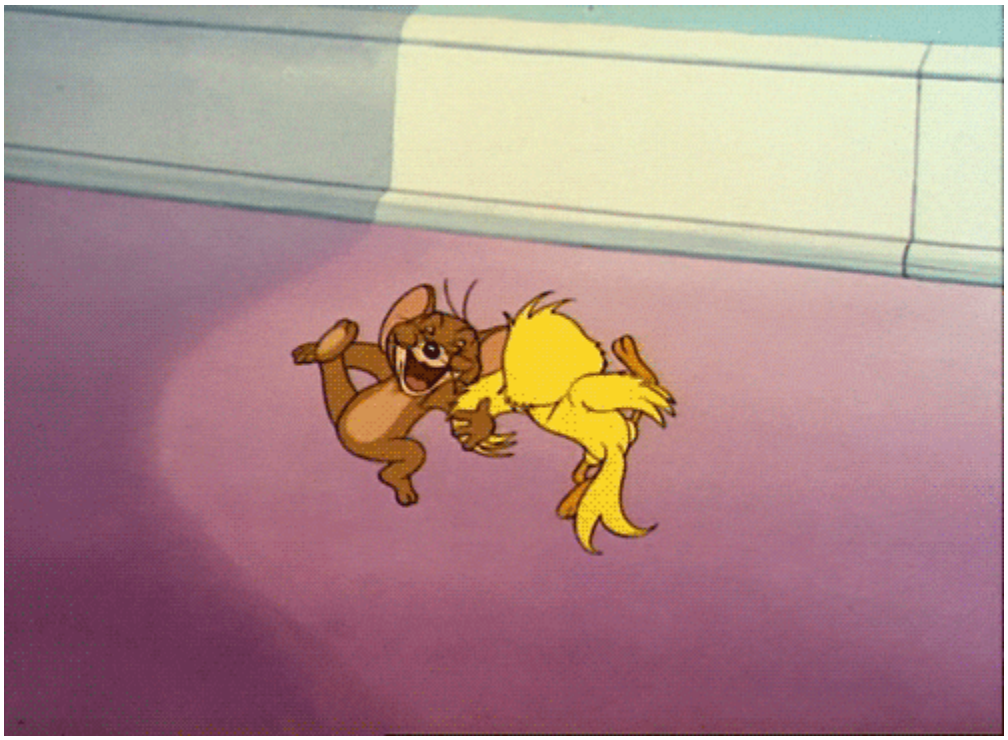
For this lab, I learned and understood how **PATCH** works. Unlike **PUT**, which updates the entire resource, **PATCH** only updates the specific fields of the resource. In this case, I used **PATCH** to update the price of the leather jacket.

Additionally, when using **OPTIONS**, it can reveal the methods that are supported and required by the endpoint.

But I have also learned two new HTTP methods: **TRACE** and **HEAD**.

- **TRACE** is used for debugging and testing communication purposes between the client and the server.
- **HEAD** only retrieves the headers of the response, not the entire content.

That’s all for today. 👋



Signing off, noob3InT 📝