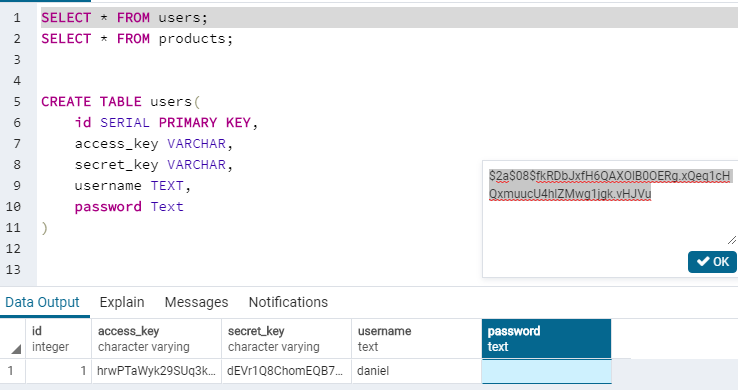
Daniel Moloney C15446648 Lab 2 submission explanations

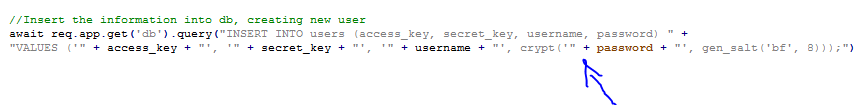
## Firstly, I would like to apologise for the late submission. Personal circumstances have been unfortunate the past few weeks. A close friend of mine’s mother went into remission for her brain cancer and my scheduling was thrown off by it. The first submission was delayed due to that. Sadly, she passed away recently and the turmoil from that event delayed my second submission. The funeral needed to be attended and I had to support my friend. I understand if late penalties apply but I wanted to give my reason for the delays.

## On a lighter note, a major delay was due to the post command not functioning how I’d intend it to. The solution was using postman (it took way too long to figure this out). Anyhow, I will run through the lab and give my work I completed for it

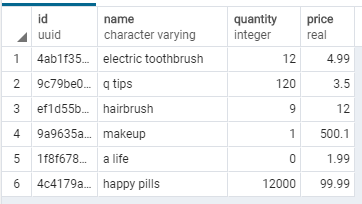
Part 1: Implement a users table having a username and hashed password fields. Use the …

I implemented this successfully. Below is a screenshot of the users table with a hashed user password and a snippet of the code hashing the password on registration.

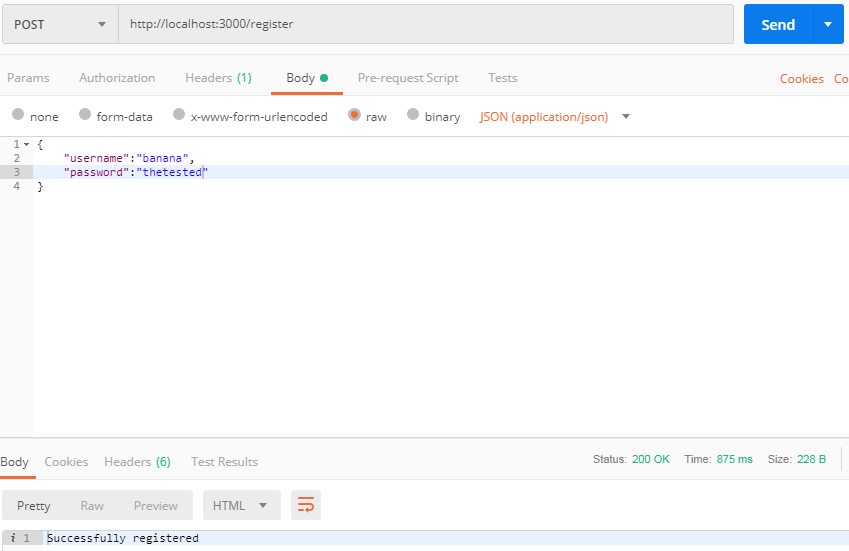




The protected resources file is a simple product table



A sample command used for registering the user would be the below in postman



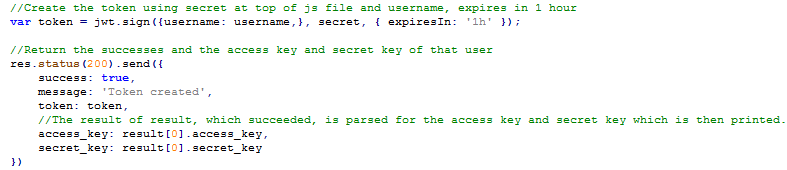
Using a general curl command, the equivalent would be:

curl -X POST <http://localhost:3000/register> -H “Content-Type: application/json” -d ‘{ “username”: “banana”, “password”: “thetested” }’

Part 2: JWT-secured access to products table. Authentication for accessing protected table

In this part I successfully created JWT-secured access.

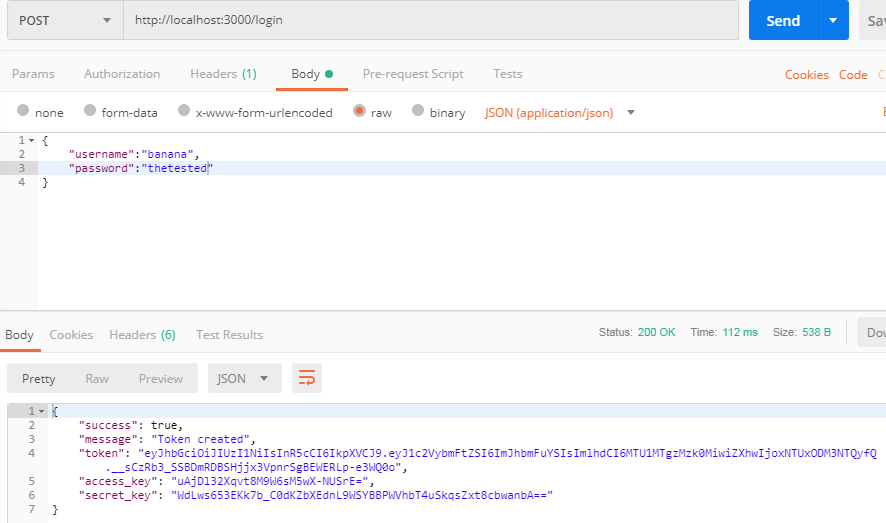
Please see below snippets of the code and functionality in place to create a web token on user login and only display products information if token is valid.



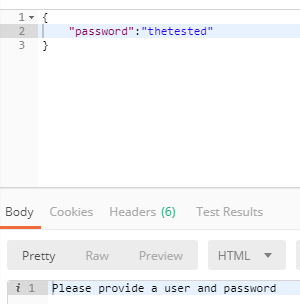
In the above code the java web token is created for the user and expires in 1 hour, the secret text as part of the signing is simple text that is provided at the top of the index.js file just for the purpose of the demonstration.

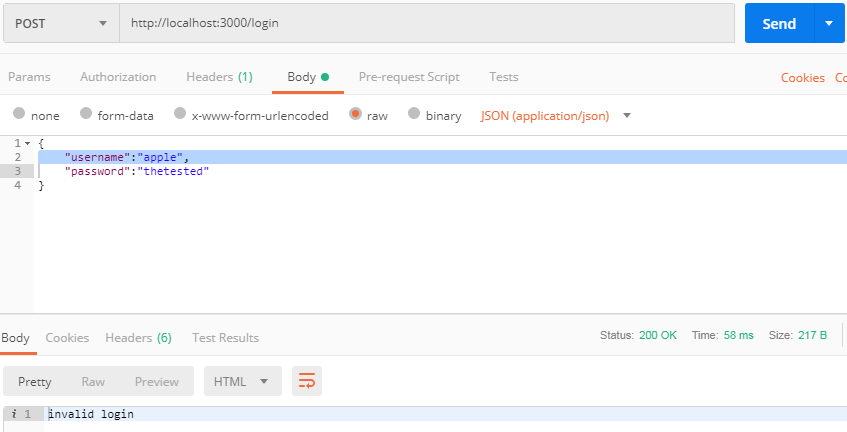


Once the token is created then the server returns the token value, as well as the users access key and secret key for later use. Successful login below.



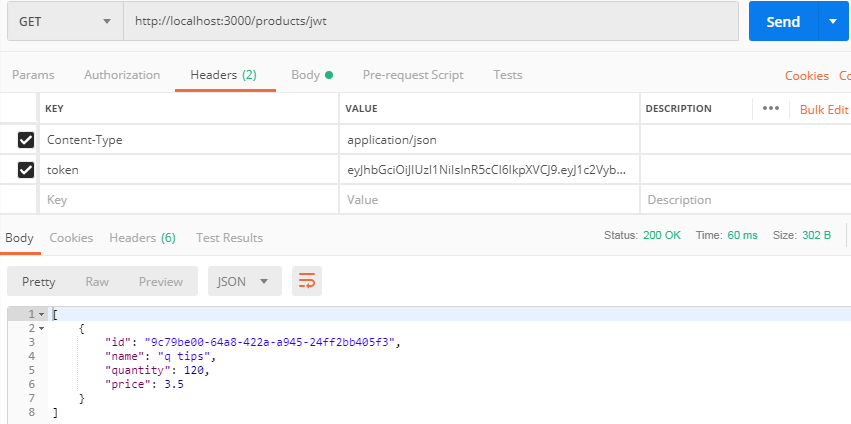
Below is the different fails

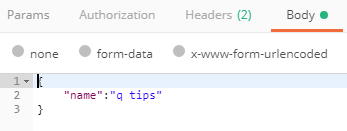


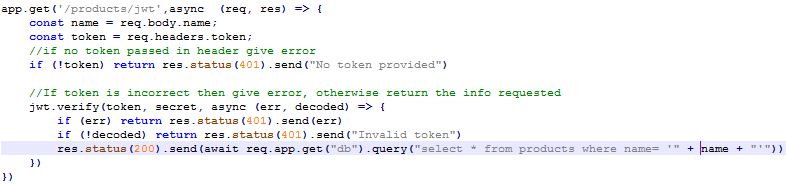


The above curl representations are almost identical to the first page example. Just change the url on the command.

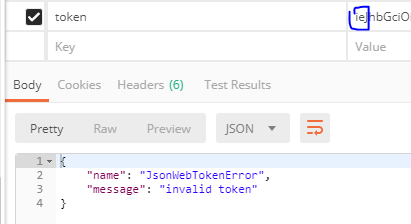
Please see below the retrieval of information from the products table with the use of valid authorization







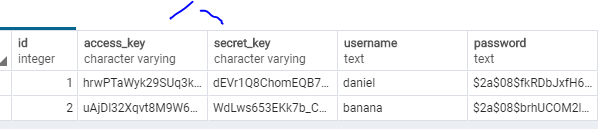
As you can see the token is verified and, if correct, the info requested is returned. Otherwise error.



Sample curl is below:

curl -X GET <http://localhost:3000/products/jwc> -H “Content-Type: application/json” -H “token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImRhbmllbCIsImlhdCI6MTU1MTgwNDkwMywiZXhwIjoxNTUxODA4NTAzfQ.E\_8dAuSoR2K6JE\_UxSPekyukGrAcaI7uYAsw2PcGAfM” -d ‘{ “name”: “q tips” }’

Part 3: Part 3 is integrated into the users table already.

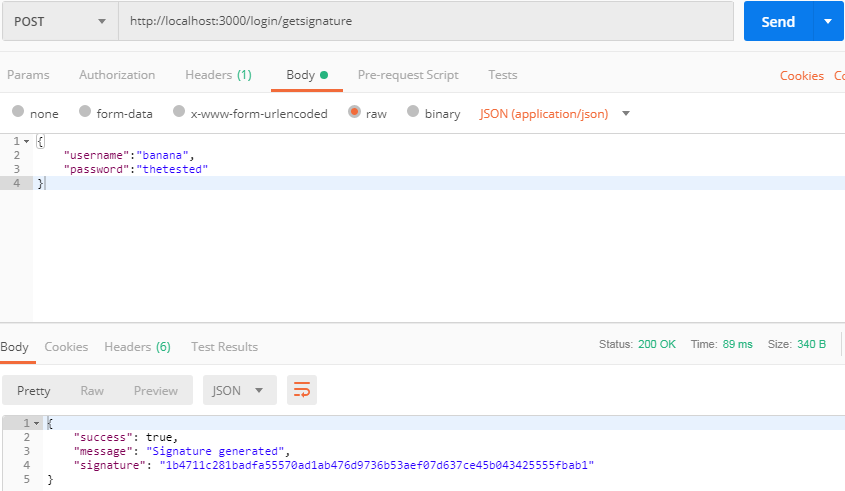


The values are inserted along with username and password automatically when registering



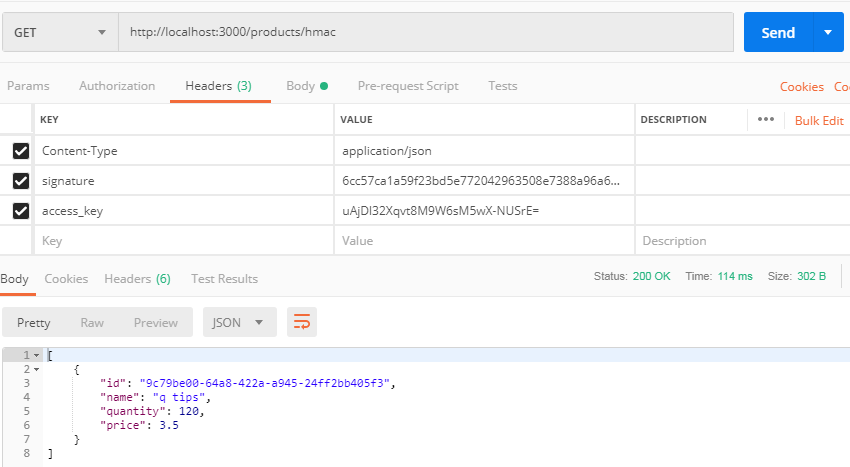
Part 4: HMAC authentication is created successfully. /login/getsignature generates a hash based on the secret key of the user (credentials are passed for verification) and the name of the product the user wishes to retrieve (in this example it is hard coded to be “q tips”).

/products/hmac is used to request the product using hmac verification. The access key and the signature are sent in the header fields. The name of the product the user wishes to request is sent as part of the body. The secret key is acquired using the public key and then a new hash is generated using the secret key and the body content (name). If the generated hash matches the signature that was sent (the signature received from /login/getsignature) then we know the data was not corrupted and the user can be authenticated.

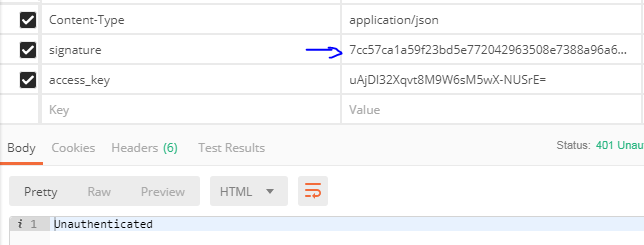


User successfully entered credentials so the hmac hash was generated. Now it is time to take the signature and use it for /products/hmac for verification before product table row is returned.

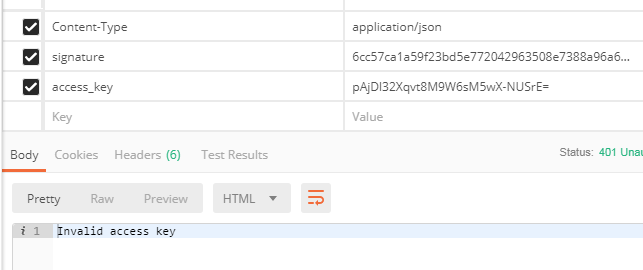
Successful retrieval of info with correct signature and access key



Incorrect signature



Incorrect access key



Curl example for the hmac authentication

curl -X GET <http://localhost:3000/products/hmac> -H “Content-Type: application/json” -H “access\_key: uAjDl32Xqvt8M9W6sM5wX-NUSrE=” -H “signature: 6cc57ca1a59f23bd5e772042963508e7388a96a65ce60ae338caf22673872d17” -d ‘ { “name”: “q tips”}’

## Thank you for reading my demonstration and, again, apologies for later submission. Life will get back on track now that the unforeseen events have ended.