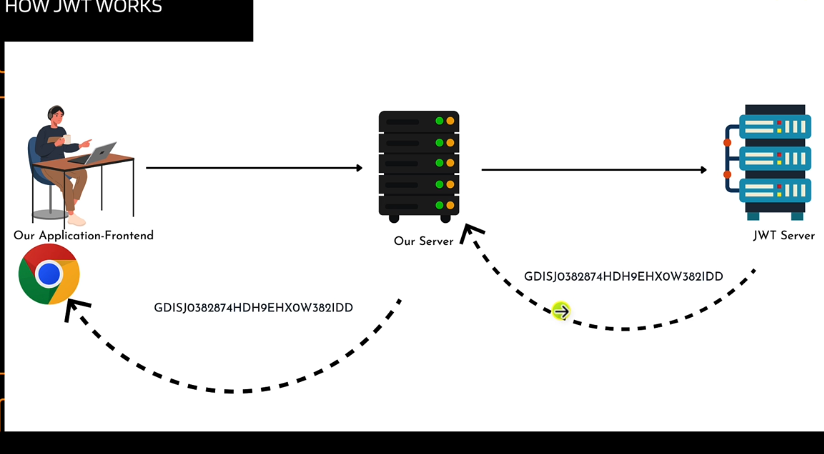
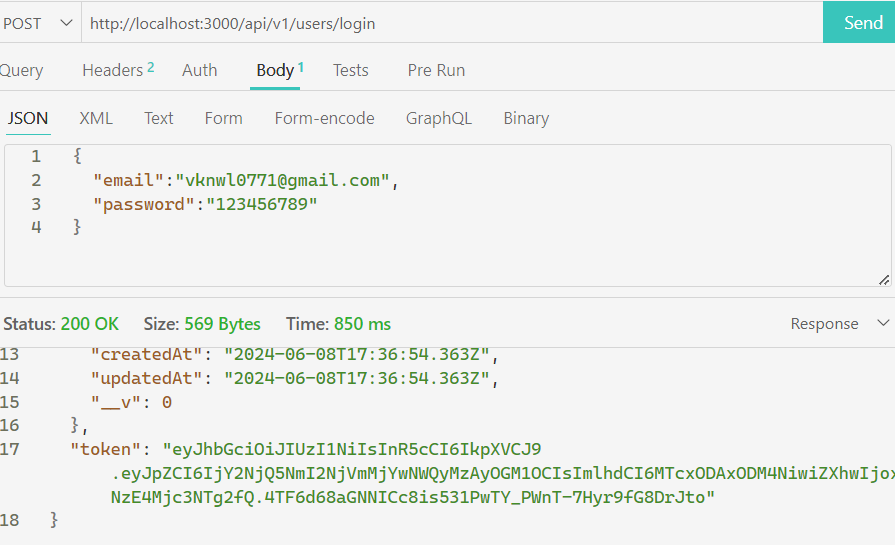
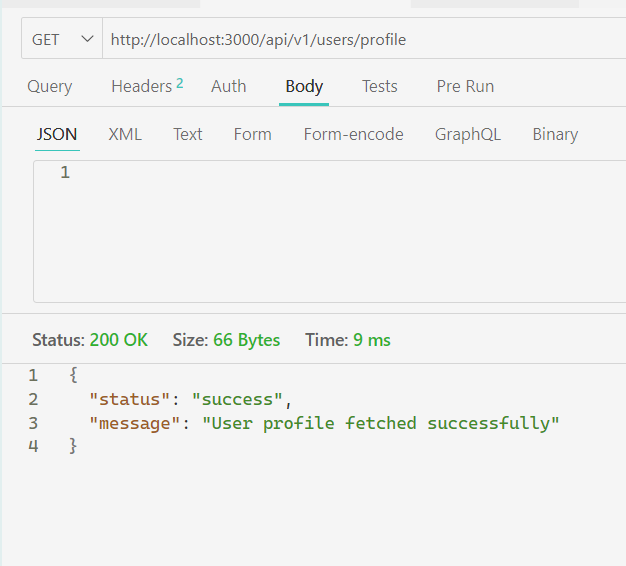
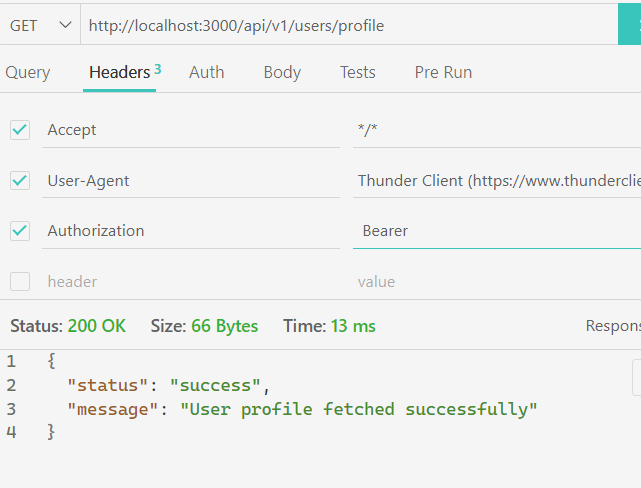
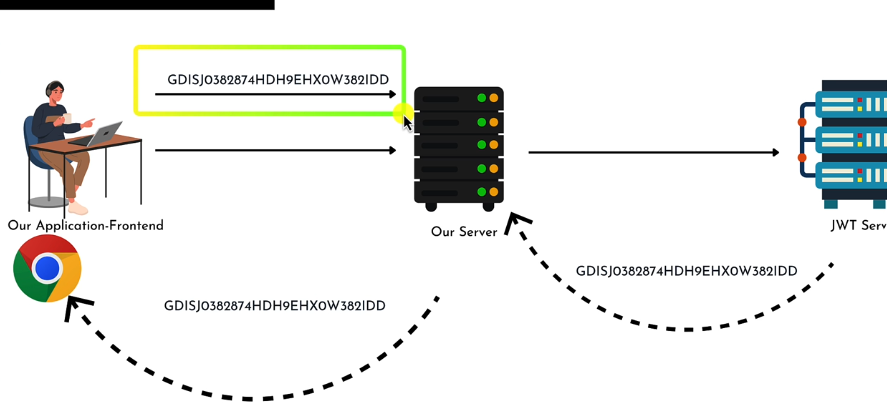
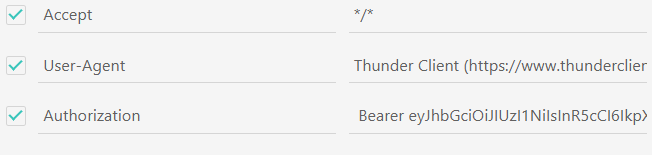
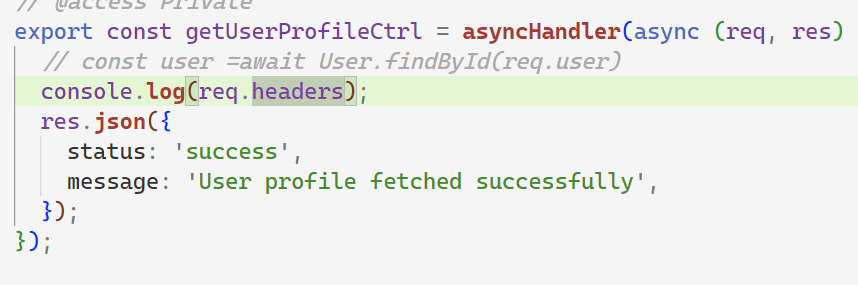
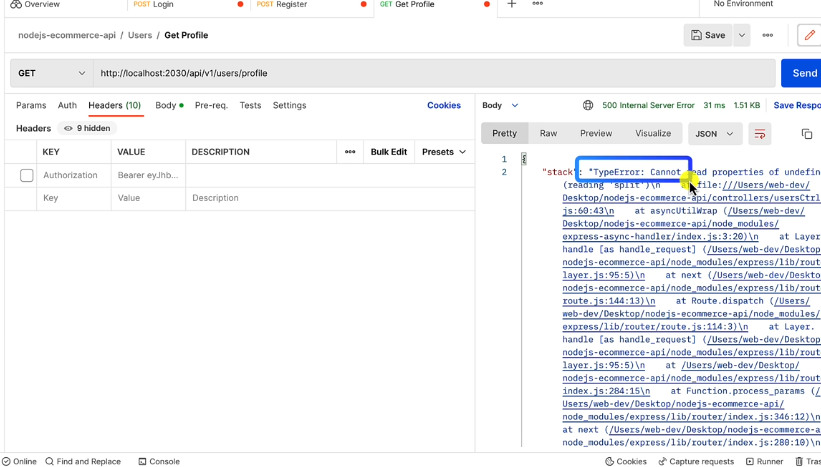
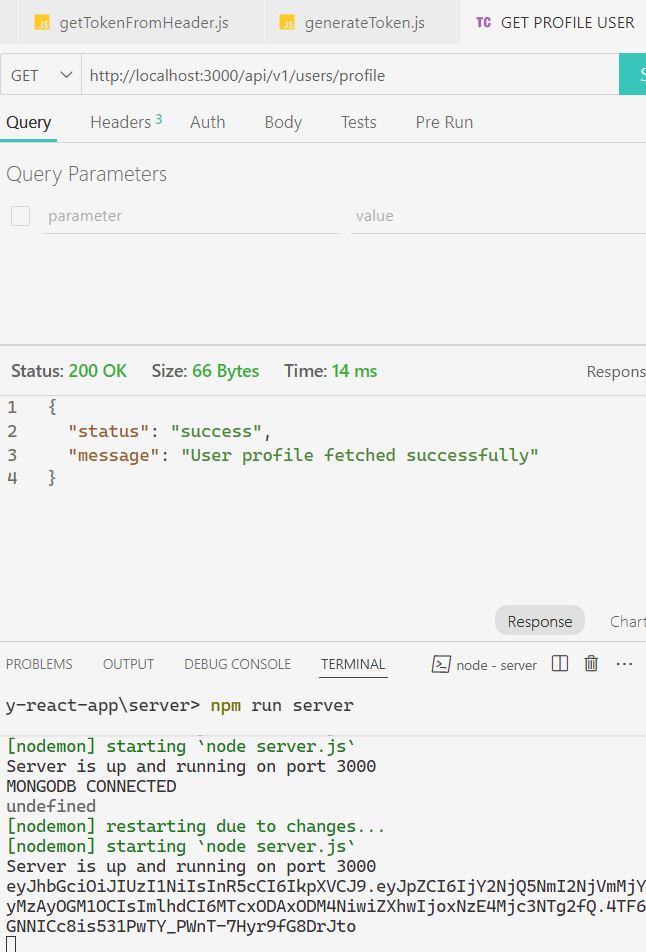
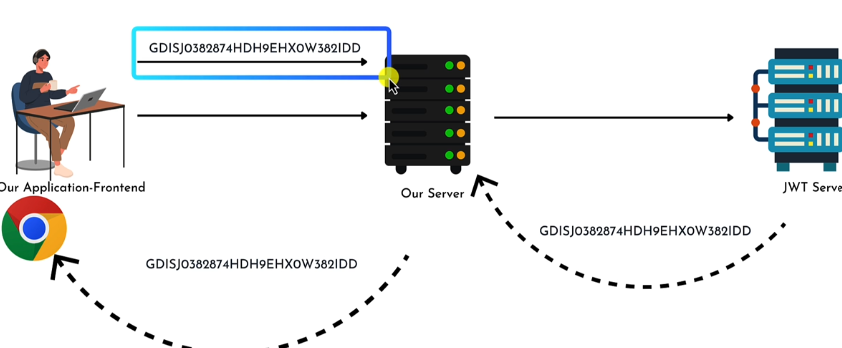
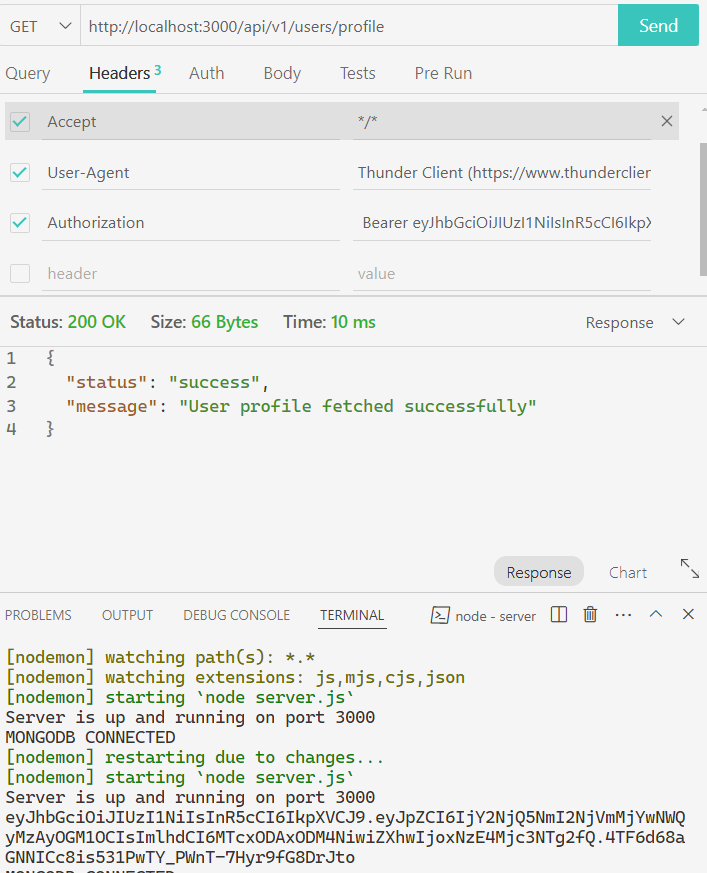
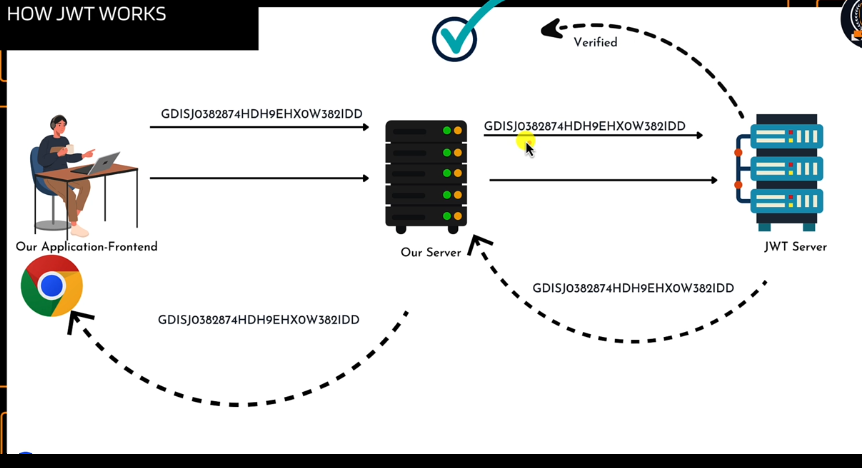
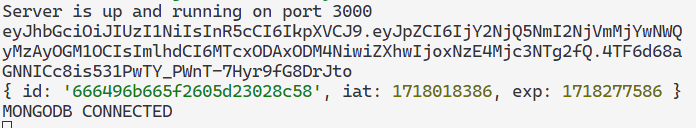
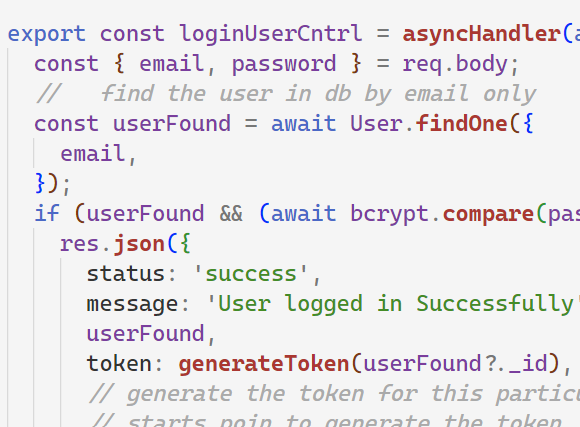
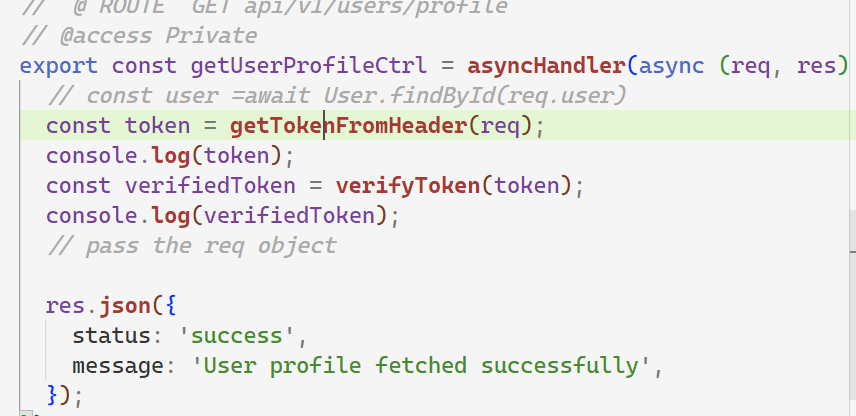
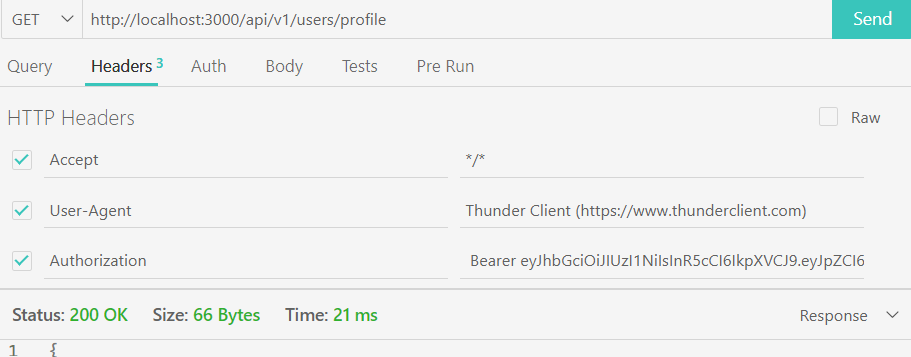
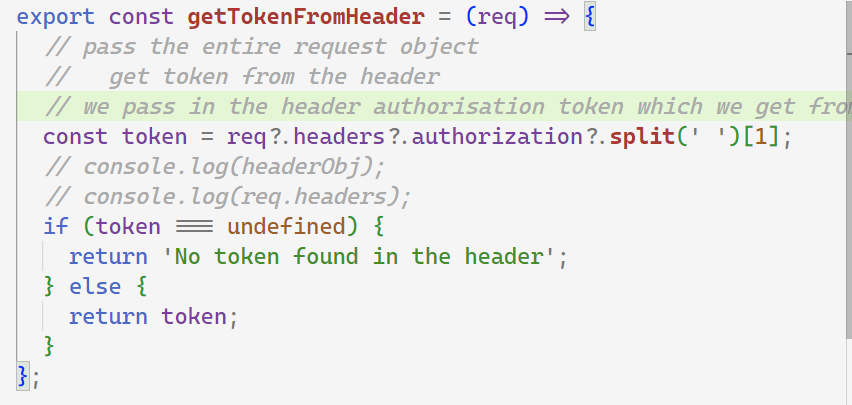
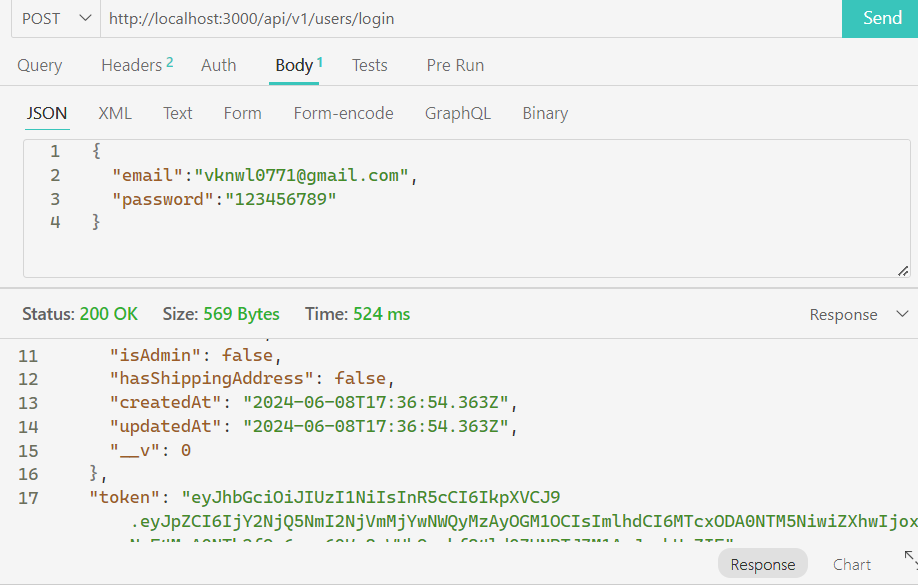
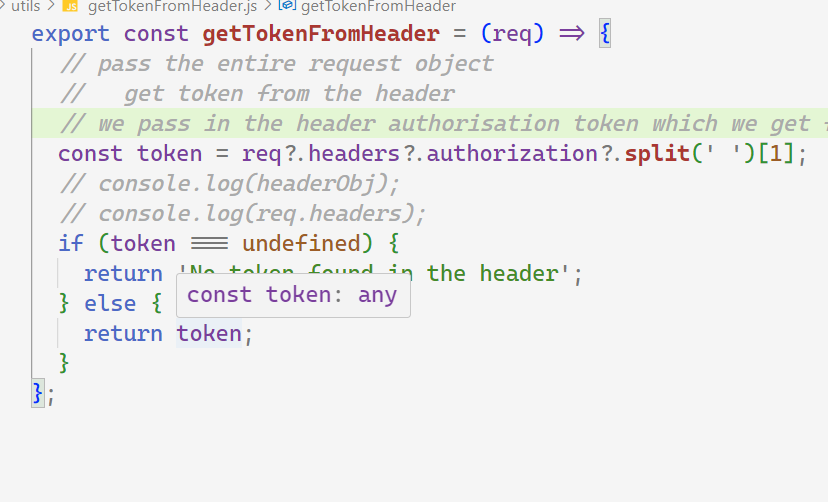
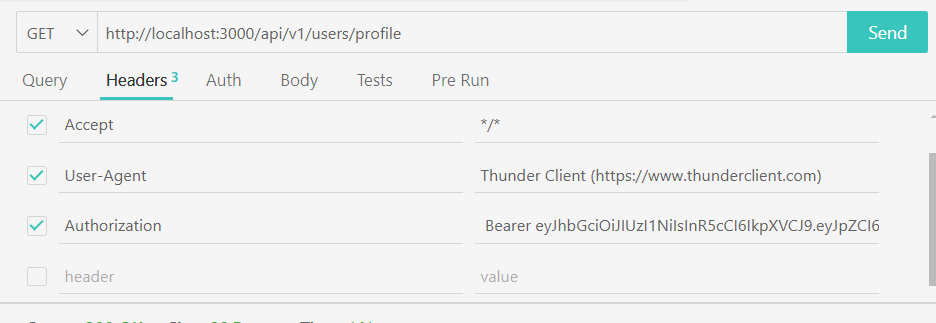
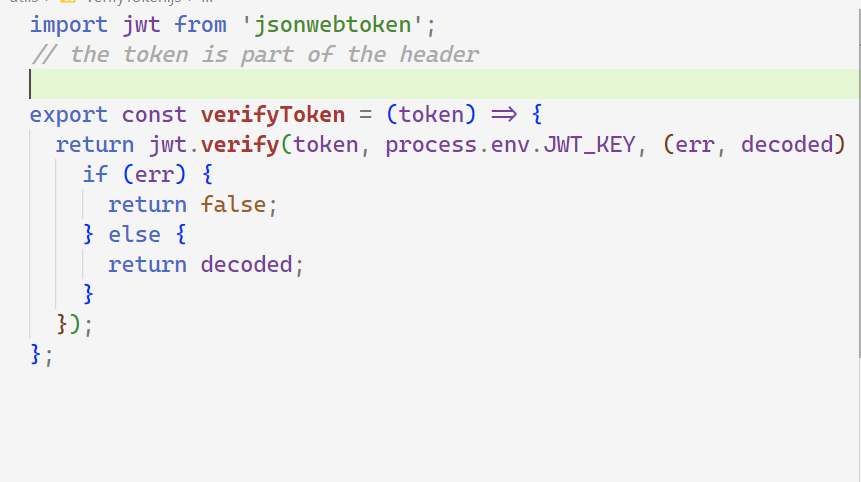
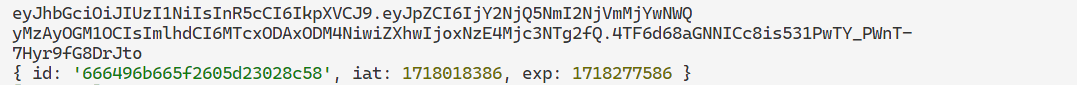
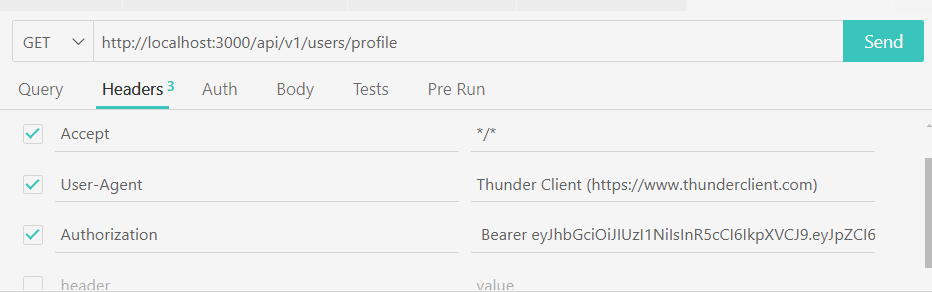
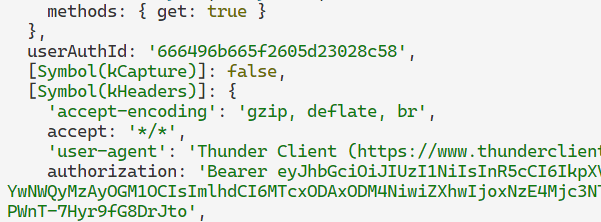
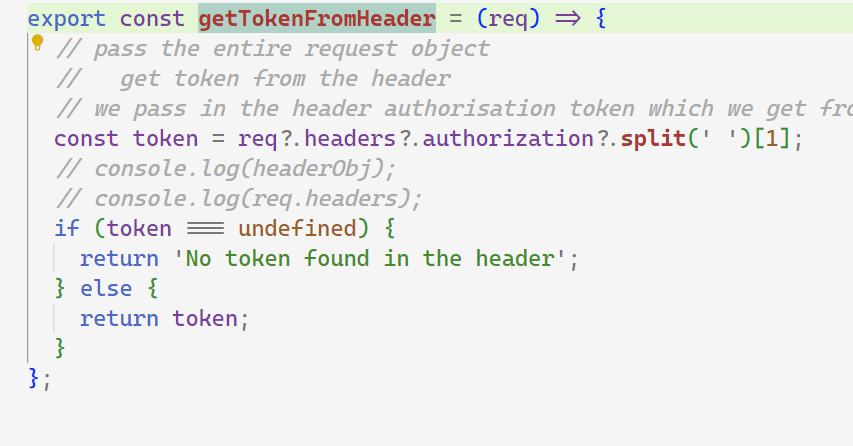
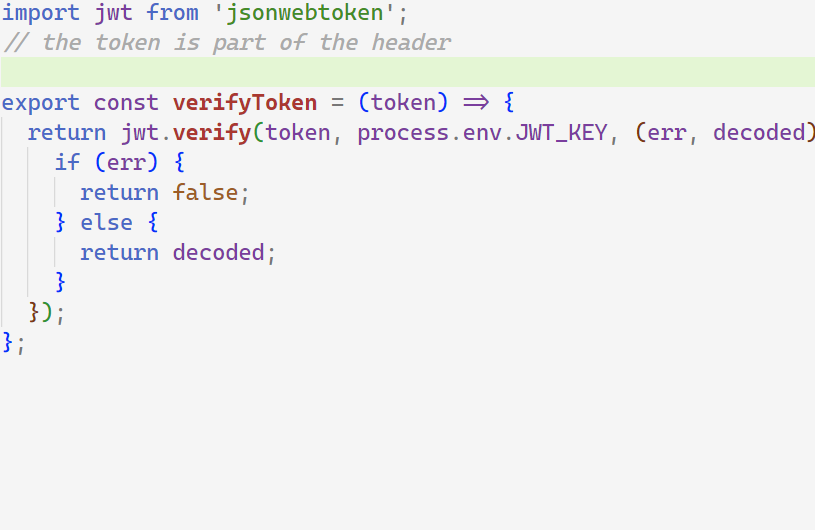
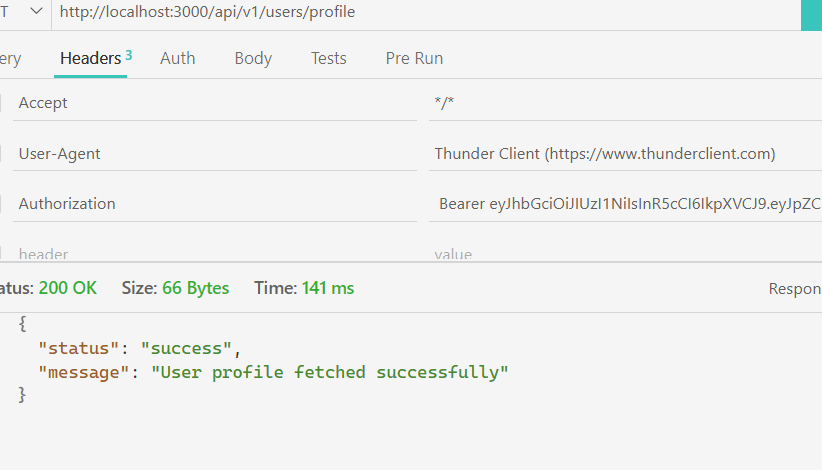
1. GENERATE TOKEN FOR LOGIN
   * + 1. Gerate the token and make the file in the utils and called generateToken and usind it in the loginUserController function
       2. 
       3. 
2. DUMMY PROFILE CONTROLLER
   * + 1. 
       2. Make the function in controllers🡺 userController.js called getProfileCntrl
       3. Which returns the profile of the specific user
       4. Change the repetition of the routes in the url /api/v1/users in the app.js
3. GET TOKEN FROM THE HEADERS
   * + 1. 
       2. 
       3. WE ARE SEND THE TOKEN AS PART OF REQUEST FROM THE CLIENT
       4. 
       5. I WANT to grab the token from the controller
       6. 
       7. 
       8. We add this property
       9. We want to grab the token from the req.headers
       10. 
       11. Now split it into the array
       12. 
       13. 
       14. Problem if the user didn’t provide the token
       15. 
       16. We use optional chaining
       17. 
       18. 
       19. 
       20. We are done this one part
       21. 
4. VERIFIED GENERATED TOKEN
   * + 1. 
       2. NOW we have to verfy the token by sending the token to the jwt and it confirms us whether the token is verified or not
       3. Make ther verifyToken.js
       4. 
       5. We got the id which we send in the loginUserCntrl function
       6. 
       7. Protect the token
5. ISLOGIN MIDDLEWARE
   * + 1. Flow
       2. When I login by the loginUserCntrl function I got the token by sending the id which the particular id is login
       3. 
       4. I want to profile then I make the 2 function in utile generateToken.js and verified token bothe are working return token is the object which is the user id which I was given in the loginUerCntrl function
       5. 
       6. To login we make the function
       7. I send the req from the profile function which get the req object
       8. 
       9. By copying the token from the login url endpoint
       10. 
       11. 
       12. Flow
           1. Make the loginUserCntrl
           2. 
           3. Then userFound has the id which we passed to the generateToken and generate the token when I login with the particular email, and password
           4. 
           5. It got the same req object takes the id and generate the token
           6. 
           7. Got the token
           8. Now in getUserProfileCntrl function we get the token from the get token from the header
           9. 
           10. We pass the token from the copy token and paste in the profile route while we hitting the url
           11. 
           12. So we parse the object and obtain the token token is simply consist the object which has id of that loginUser which we set in the loginUserCOntroller
           13. Then we verified the token
           14. 
           15. Returen the decoded
           16. 
           17. Now we check the islogin while making the function isLoggedin.js
           18. 
           19. isLoggedIn is the middleware when hit the endpoint it get the req object which we pass during the hitting
           20. 
           21. It got the token
           22. Now we get the 3 steps
           23. 
           24. We make the new field of req.useAuthId which has the decoded?.id which is from the object
           25. 
           26. This is simply taking the token from thunderclient in split manner and then decode it in the orginal form back in loginUserContrl. Function which we send the user id of login from the database
           27. So when we hitting the profile endpoint the first run the middleware it get the token fdrom the header using authorisatiion bearer token from the function getTokenFromHeade.js and it verified the toke which return the decoded form verifiedToken.js and then we make the new field in req object userAuthId which store the id and then the userControler.js has getProfile function it runs and pass the req object wchi has same req object
           28. Because first run the middleware and pass it next()
           29. 
           30. 
           31. 
           32. 
           33. 