目录

[1 Controller’s Problems 1](#_Toc1507)

[2 Add Viewer 1](#_Toc20037)

[3 link views to controller 4](#_Toc12784)

[4 test form data from the login page 7](#_Toc17585)

[5 test JWT token with postman 9](#_Toc13975)

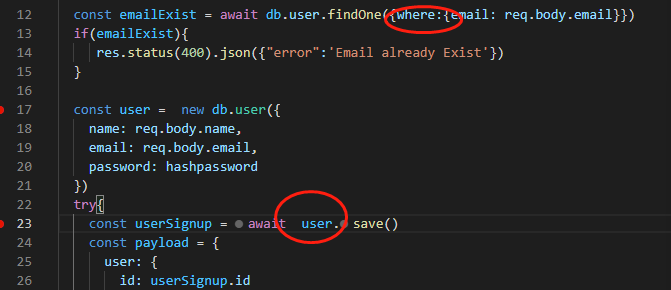
[6 add auth to all routes except login page 10](#_Toc30744)

[7 connect login frontpage with api 10](#_Toc27883)

[8. add fileupload function: routes and controller 12](#_Toc9630)

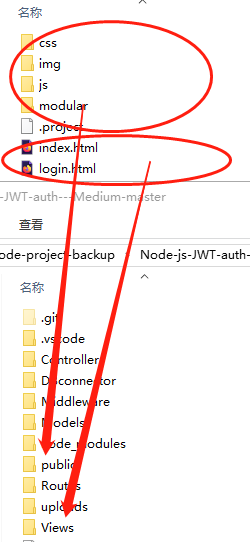
[9. Node upload to IPFS 12](#_Toc11381)

# 1 Controller’s Problems

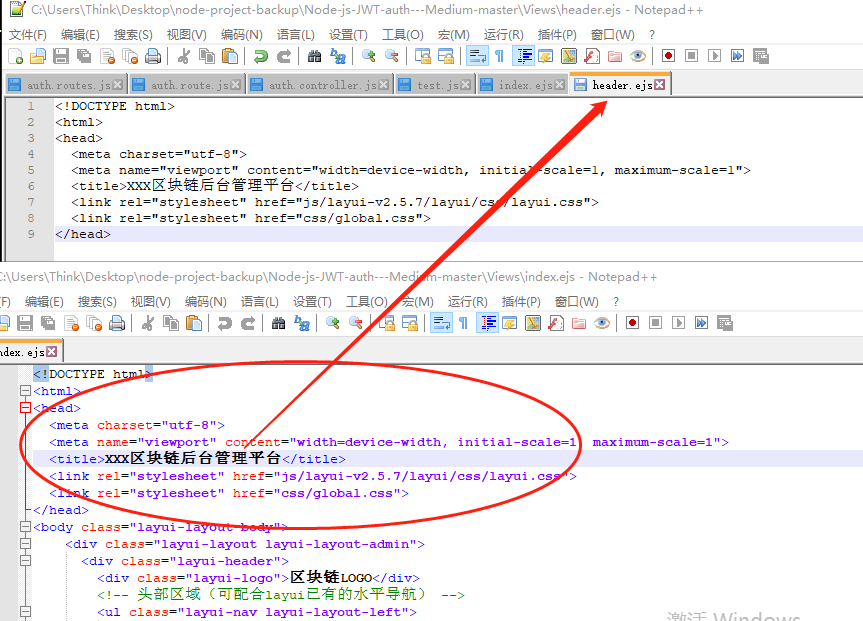


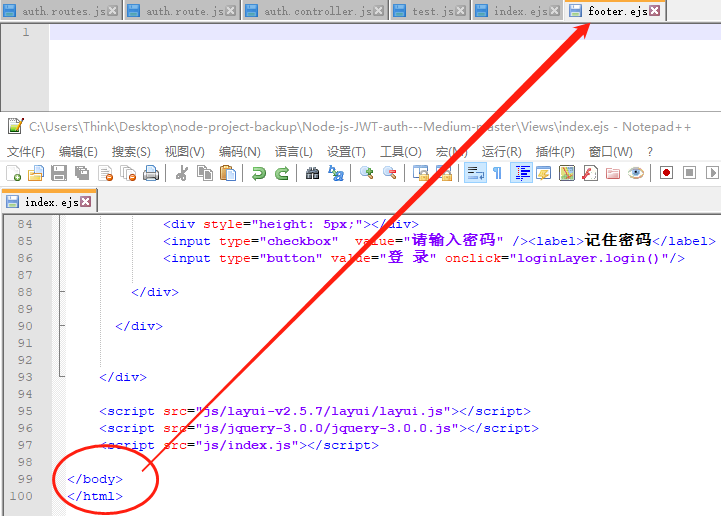
# 2 Add Viewer

1. Move css js to public folder, and change html to ejs and move them to the folder

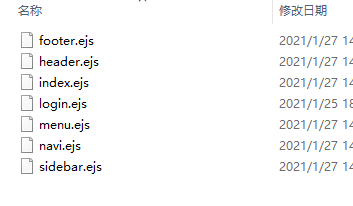


2 split html into parts

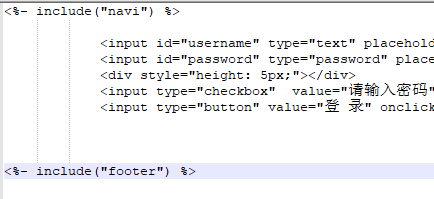




Eventually:



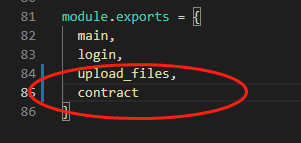
Index.ejs

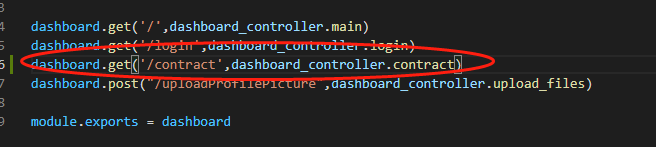


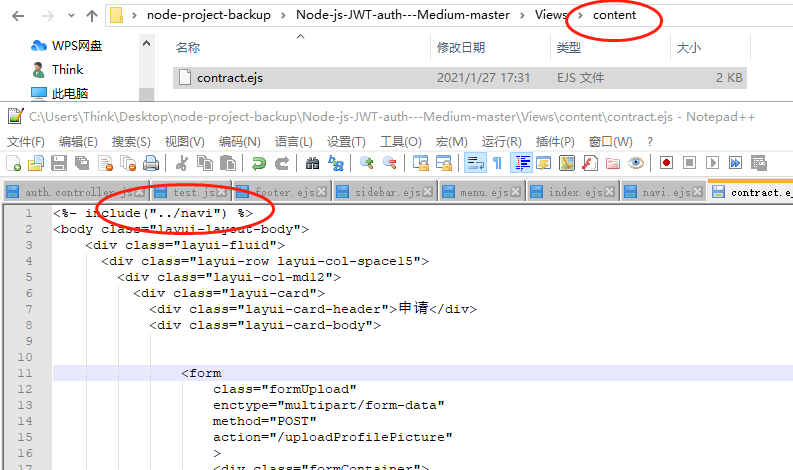
# 3 link views to controller

1. Add controller to apply\_for

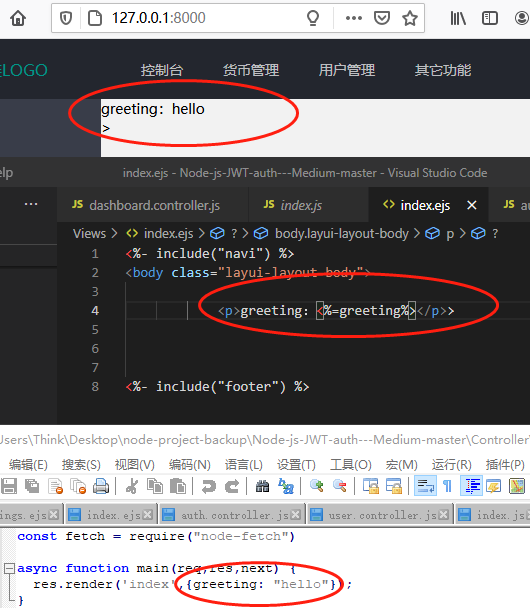






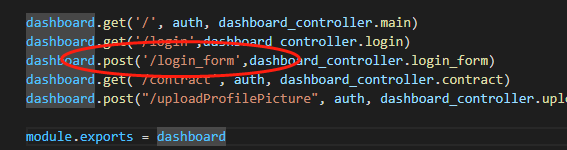


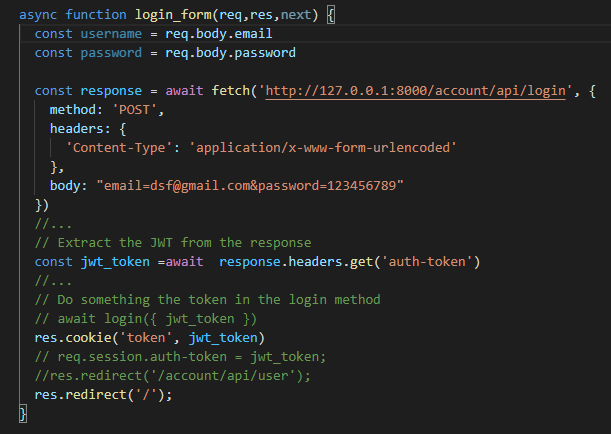
2 pass data from controller to the views



# 4 test form data from the login page

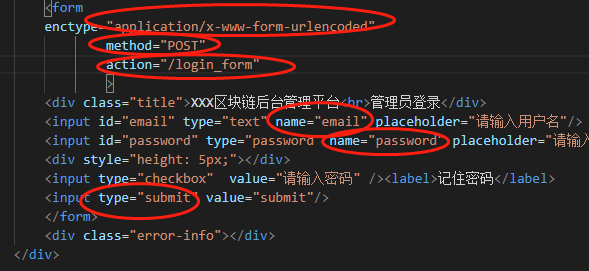
1 create a route to handle the form data, and create its corresponding controller



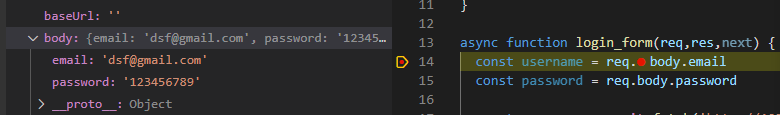


1. Create form

Set form data to www-form-urlencode, set names and set type to be submit, set action to be the route

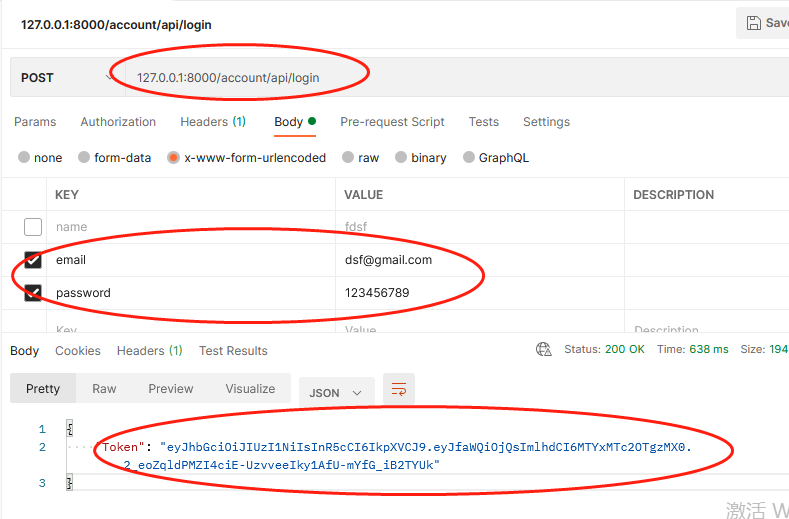


1. Test

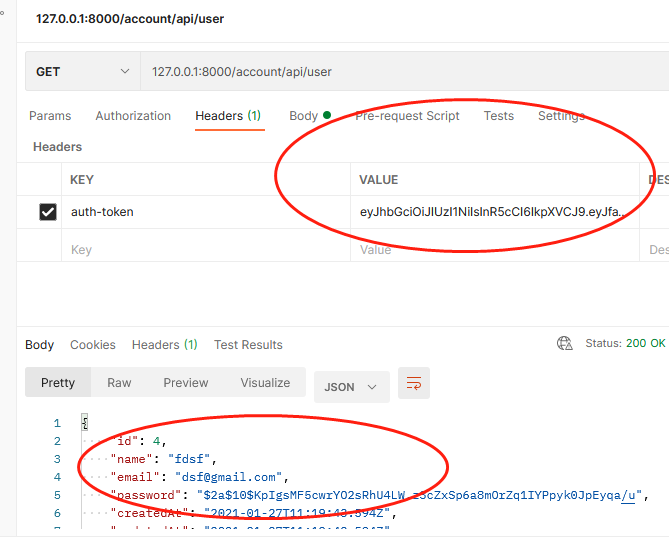


# 5 test JWT token with postman

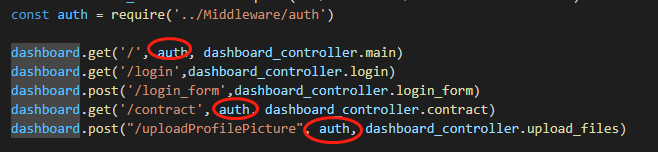
1. Get token



1. Add this token to get header

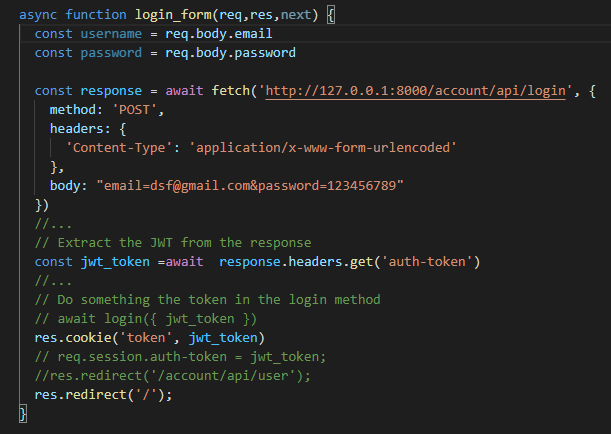


# 6 add auth to all routes except login page



# 7 connect login frontpage with api

1 add login logic



2 use cookie to store token



1. Test

It will return user info in json format on the browser

# add fileupload function: routes and controller

# Node upload to IPFS