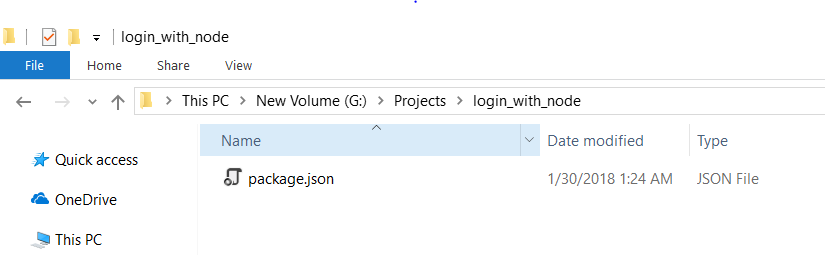
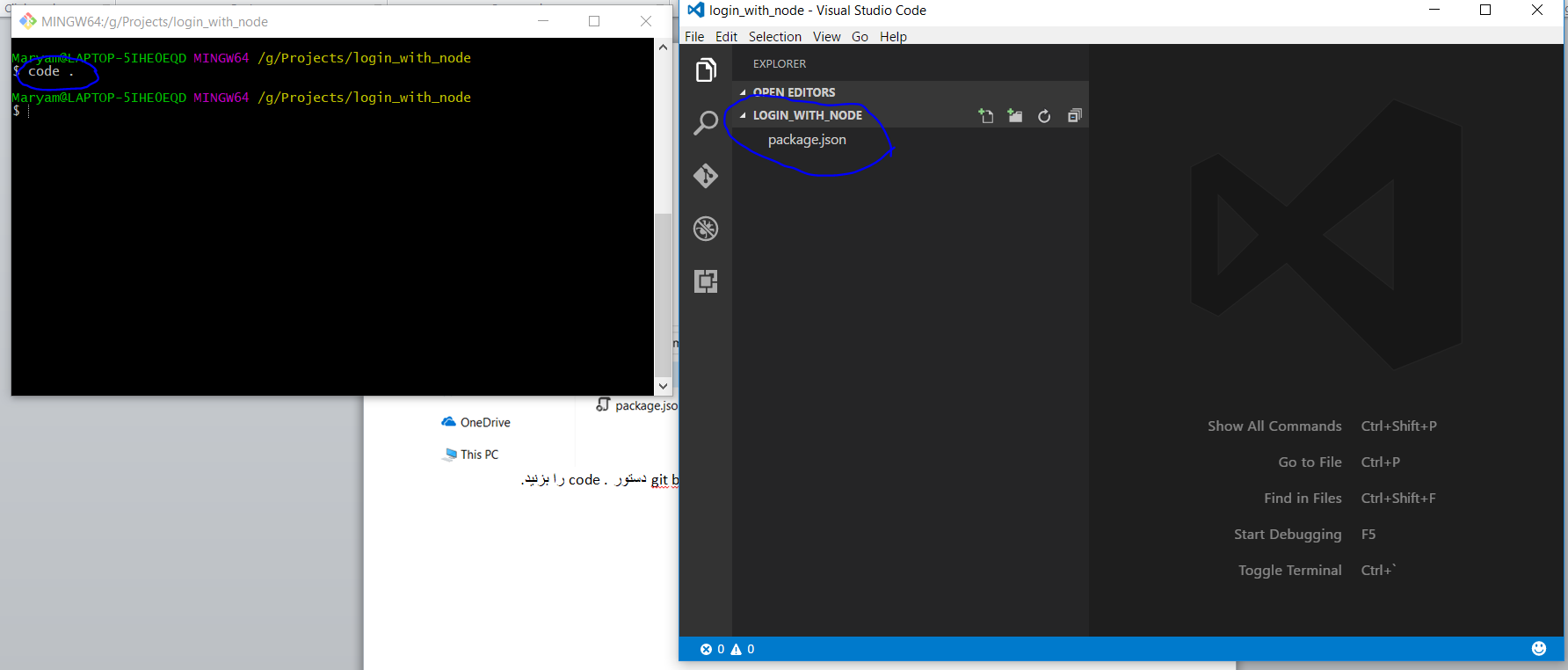
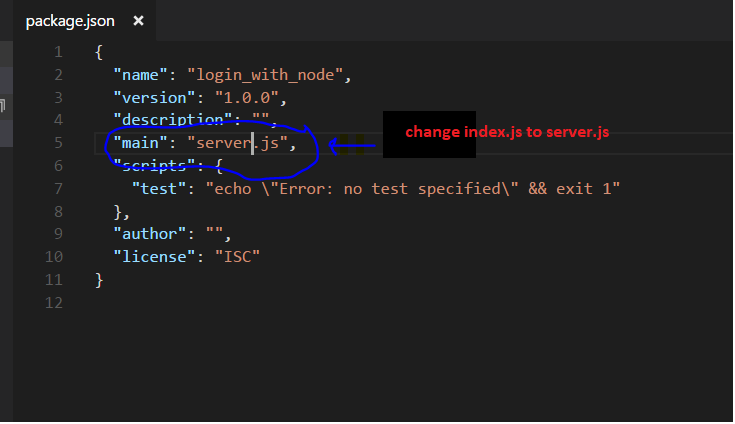
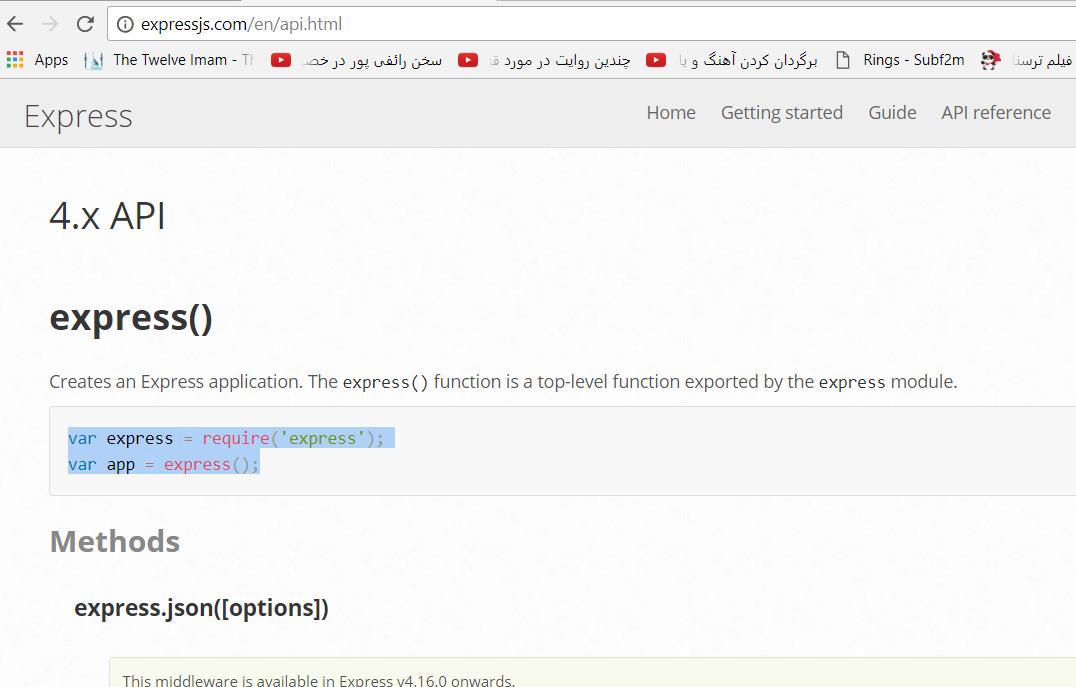
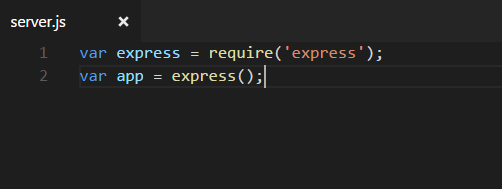
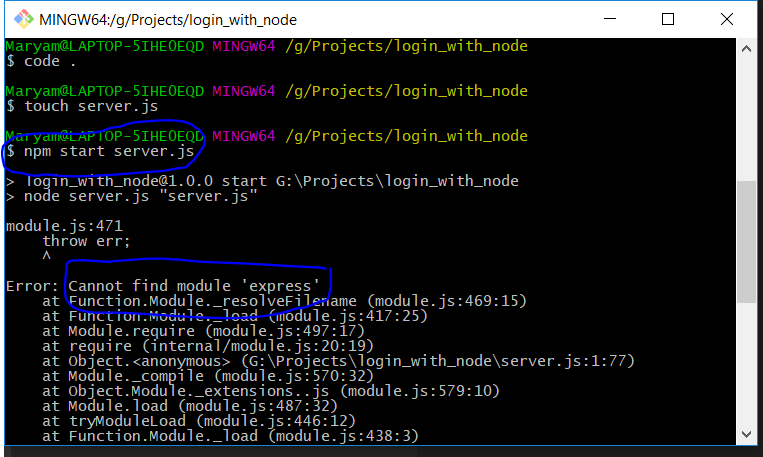
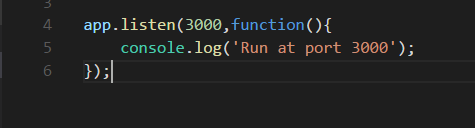
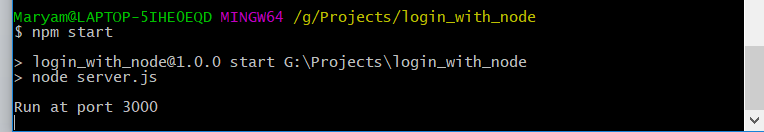
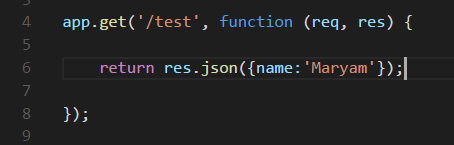
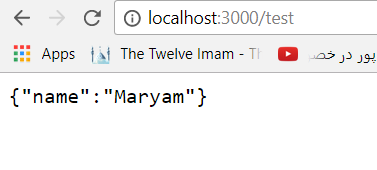
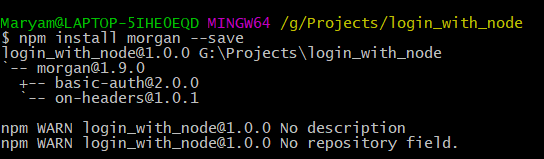
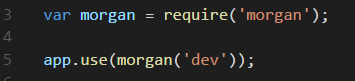
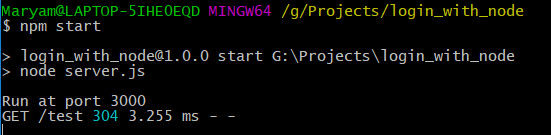
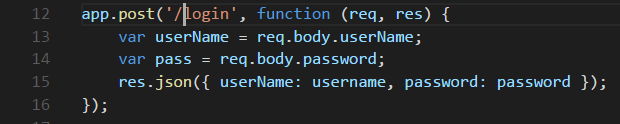
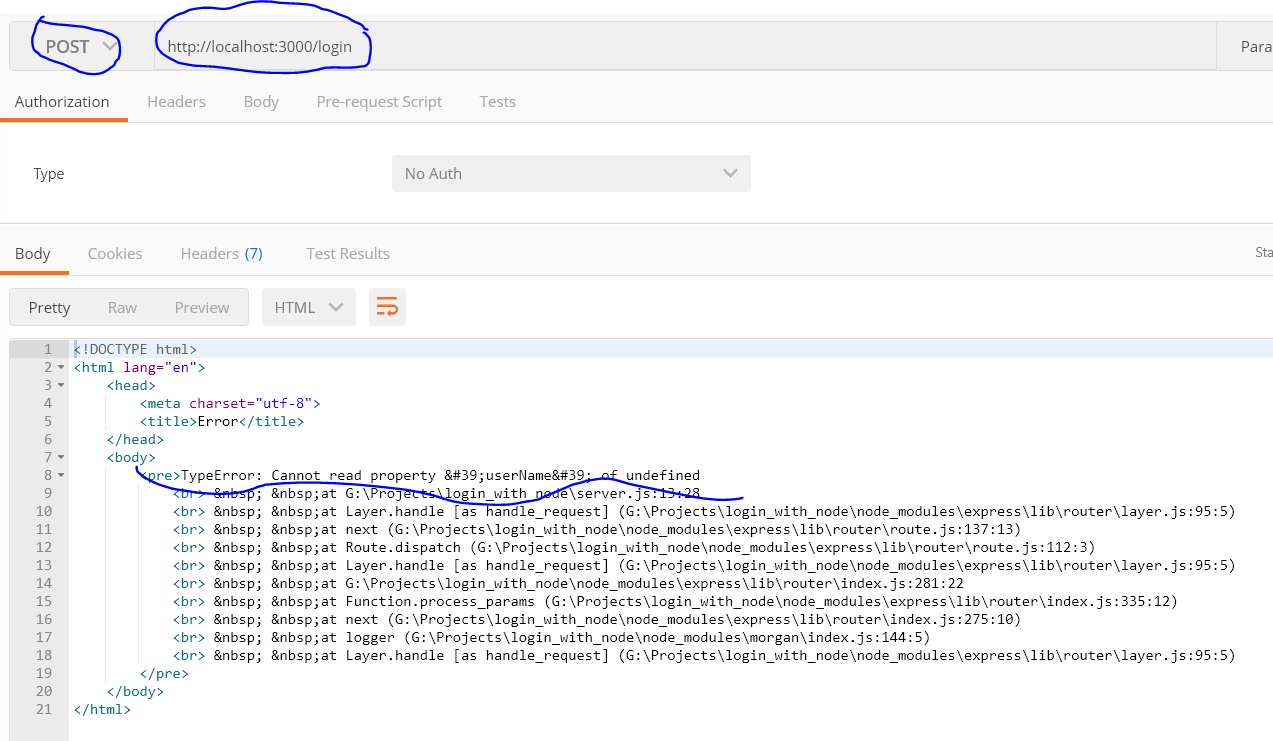
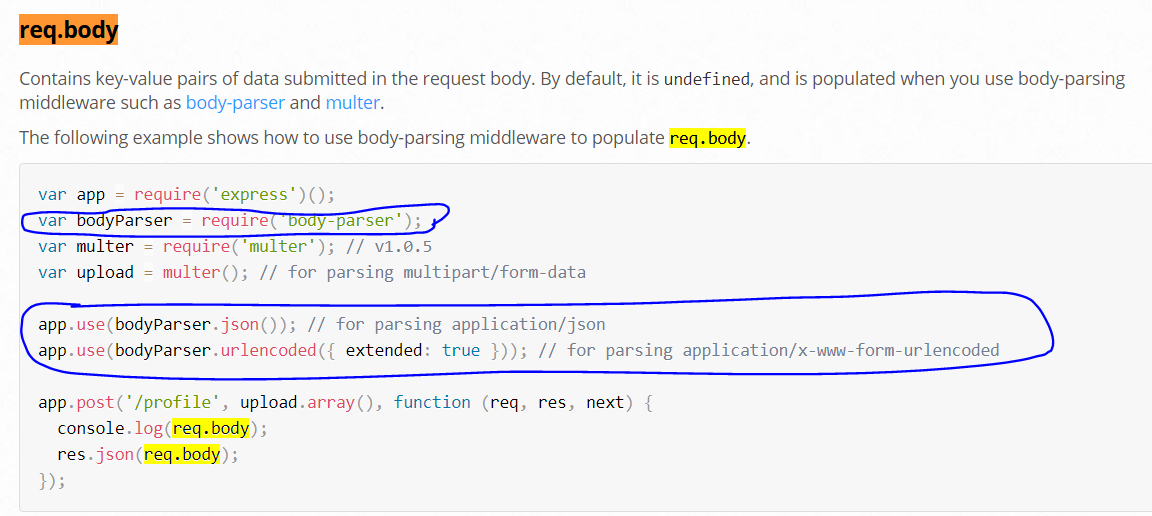
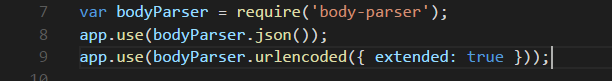
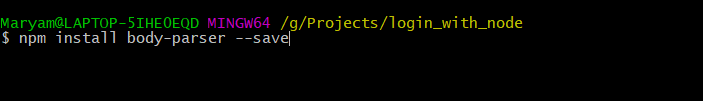
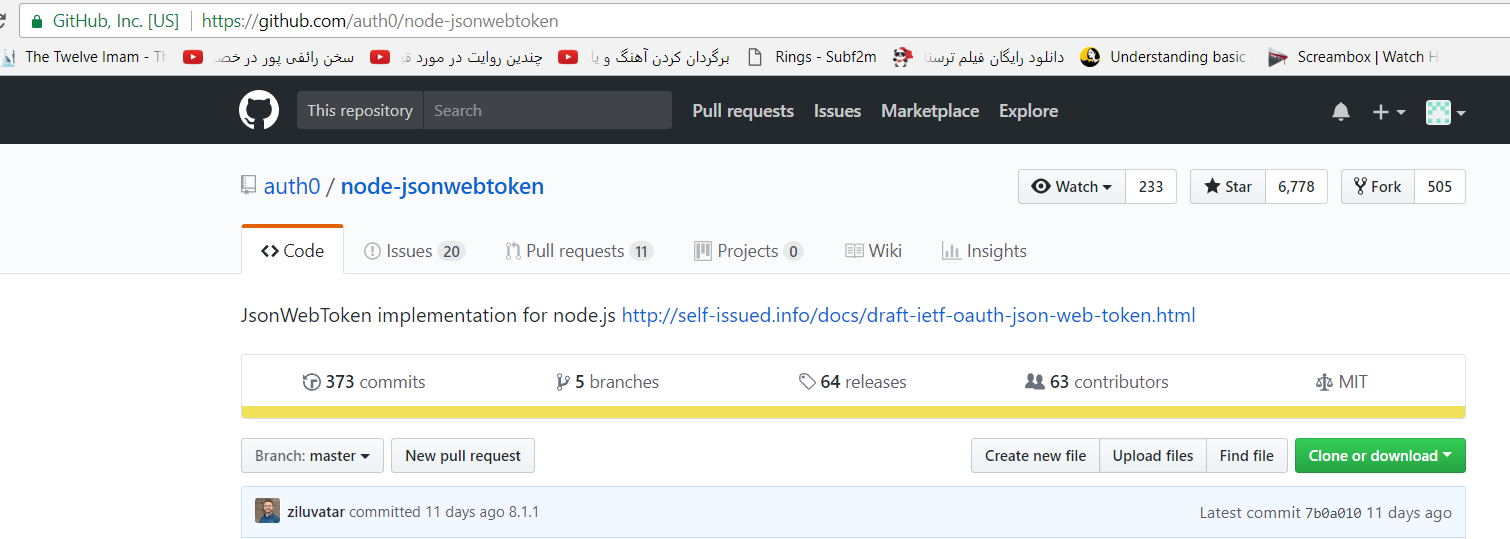
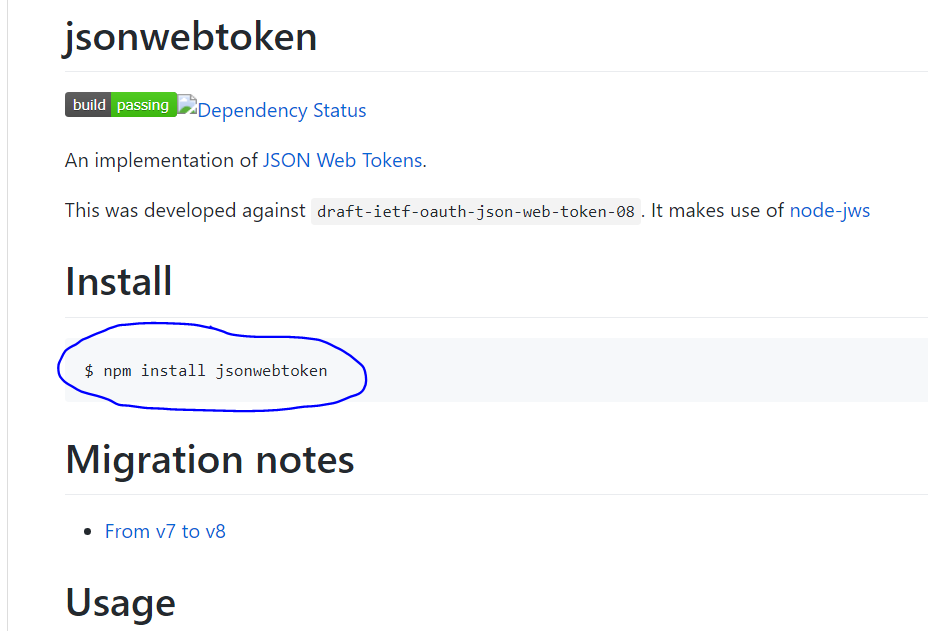
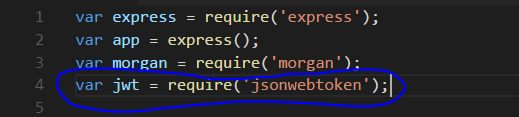
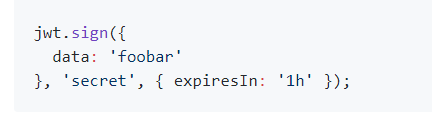
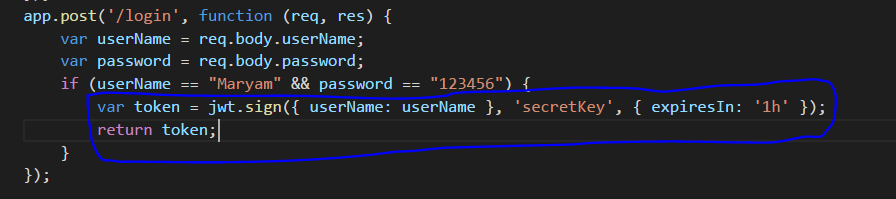
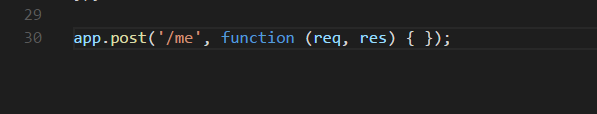
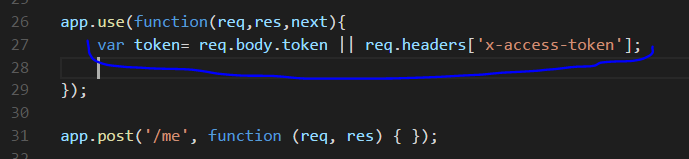
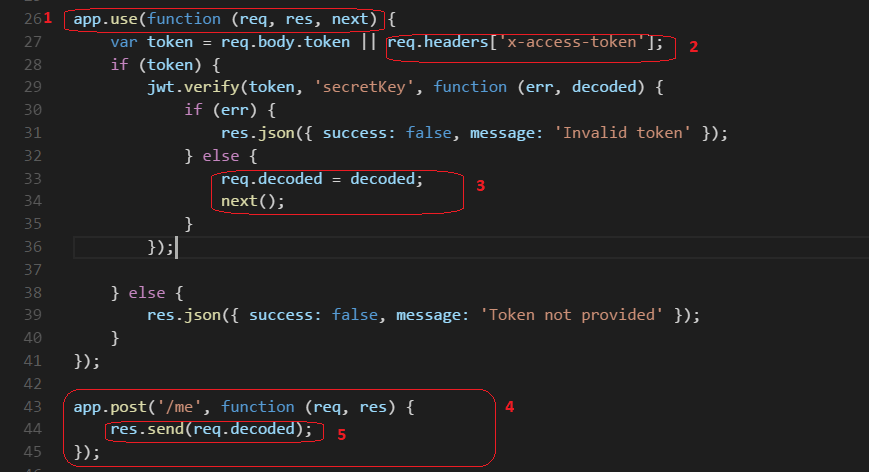
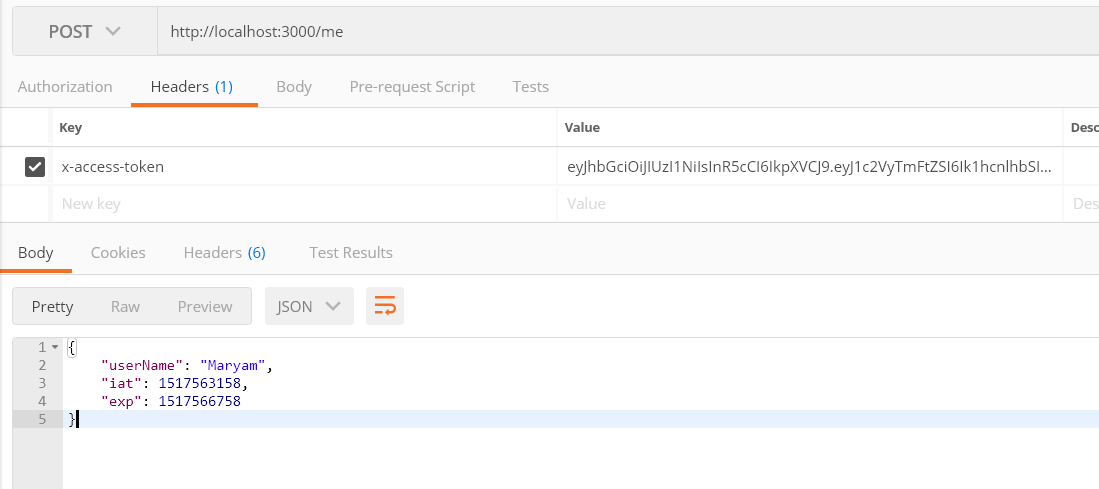
1. يك پوشه به نام login\_with\_node ، ایجاد کنید.
2. Git bash را باز نموده و به داخل آن redirect نماييد
3. با دستور node –v مطمئن شوید که Node نصب است (nodejs.org)
4. با دستور npm –v مطمئن شوید که npm نصب است
5. با دستور npm init ، structure لازم برای Rest Api را ایجاد نمایید.
6. 
7. پوشه بررنامه را در visual studio coed باز نمایید .برای این کار در git bach دستور code . را بزنید.
8. 
9. با command ، touch server.js ، فایل مورد نظرتان را ایجااد نمایید.
10. 
11. فایل package.json را باز نمایید . می بینیم که هیج dependency وجود ندارد ، در ابتدا دو module را لازم داریم نصب نماییم
12. به سایت expressjs.com بروید و یا google کنید و سایت را پیدا نمایید .به URL زیر <http://expressjs.com/en/api.html> ، navigate کنید .
13. دو خط اول را که برای instance سازی از express module ، هست را در فایل server.js ، past نمایید .
14. 
15. 
16. الان اگر به git bash برویم و بخ.اهیم برنامه را run کنیم ، error میگیریم ، چون از modul ی استفاده می کنیم که هنوز install نشده .
17. 
18. برای نص کافی ایت که به صورت زیر عمل نماییم .
19. در git bash ، تایپ می کنیم **npm install express –save**
20. با این کار هم module نصب می شود و هم package.json به روز می شود .
21. در ادامه خط گد زیر را اضافه می کنیم .
22. 
23. و برنامه را run مب کنبم .
24. 
25. برای تست ساده یک route به API اضافه می کنیم .
26. 
27. در ادامه برای تست ، هم URL ، <http://localhost:3000/test> را در browser بزنید :
28. 
29. ماژول morgan را هم نصب می کنیم برای log انداختن طرف server .
30. 
31. در کد تغییرات زیر را انجام می دهیم .قبل از تعریف route ها باید این کار را انجام دهیم .
32. 
33. الان اگردر browser ، localhost/test را refresh نماییم ، باید یک خط log در console ببینیم.
34. 
35. در ادامه یک action به نام login به service اضافه می کنیم وusername و password را به آن pass می دهیم و کد آن به صورت زیراست .
36. 
37. همانطور که می بینیم هیچ گونه security در حال حاضز وجود ندارد و ما تنها دو مقدار را از req.body می خوانیم .
38. برای تست کد بالا postman را باز نمودخ و به URL ، <http://localhost:3000/login> ، و متد post ، redirect ی کنیم .
39. Error ی که می کیریم به این صورت است که نمی تواند username را از مقدار null بخواند .
40. 
41. برای رفع این مشکل و برای اینکه express بتواند با post هم کار کند باید middle ware ی به نام bodyParser را به برنلمه اضافه نماییم .
42. برای اینکه متوجه شویم که چگونه باید این کار را انجام دهیم ، به سایت exressjs.com می رویم.
43. 
44. کد زیر را بعد از تعریف نمودن app ، و قبل از تعاریف مربوط به rout ها به برنامه وارد می نماییم.
45. 
46. برای نصب bodyParser ، در git bash ، command ریز را وارد می کنیم . دقت کنید که اکر در subfolderی از برنامه هستید ، به طور مثال فولدری دارید برای نگهدلری model ها و یا فولدری برای نکهداری route controller ها ، ابتدا به main root رنامه گه همان کنار فایل pacjage.json و server.js است navigate کنید در git bash و سپس command را run نمایید .
47. 
48. در ادامه اگر برنامه را دوباره run نمایید و مقادیر username و password را هم به صورت json از postman ارسال نموده و content-type را هم برابر با application/json قرار دهید ، باید بتوانیم این مقادیر را دریافت نماییم .
49. برای login می توانیم از روش تولید Token استفاده کنیم و action های مورد نظرمان را با آن Token ، secure نماییم .
50. برای اینکار در گوگل search می کنیم node jsonwebtoken و به صفحه github آن redirect می کنیم .
51. 
52. داخل وب سایت scroll down می کنیم و نام اصلی module را همراه با command ، npm install پیدا می کنیم .
53. 
54. در ادامه در git bash دستور بالا را می زنیم بعلاوه –Save
55. سپس به سایت باز می گردیم و از قسمت sample ، آن استفاده می نماییم . مثلا برای require نمودن در برنامه .
56. 
57. 
58. برای تولید Token ، از نمونه ای که rexpire دازد استفاده می کنیم که در همان صفحه ، می توانید ببینید .
59. 
60. برای برنامه تعییراتی که در کد بالا می دهیم به شرح زیر در متد login اضافه می نماییم .
61. 
62. برنامه را در postman ، می کنیم تا نتیجه را مشاهده نماییم .
63. برای secure نمودن یک action از یک service ، می توان به دو روش عمل کرد . یکی اینکه از passport module استاده نماییم و دیگر اینکه خودمان قبل از متدی که می خواهیم Secure باشد یک middle ware تعریف نماییم .
64. به طور مثال متدی یا action ی به نام ‘/me’ تعریف می کنیم و می خواهیم که آنرا secure نماییم.
65. 
66. برای تعریف middle ware ، از app.use استفاده می نماییم و در این مورد، مقدار token را در این midle ware ، چه از طریق header ، پاس داده شود و یا از طریق body و یا از طریق queryString ، آنرا دز middle ware ، بازیابی می کنیم .
67. 
68. در ادامه ، به سایت jsonwebtoken برمی گردیم تا نحوه validate نمودن token را از آنجا گپی کنیم .
69. 
70. و برای استفاده ، تعییرات مورد نظرمان را در آن اعمال می نماییم .
71. 
72. در تصویر بالا موارد زیر قابل توجه است .
    1. شماره 1) middle ware باید قبل از متدهایی تعریف شوند که می خواهیم middle ware روی آنها اعمال شود .
    2. شماره 2) زمانی که URL ، <http://localhost:3000/me> را صدا می زنیم ، در ابتدا خط شماره 1 اچرا می شود و token را از body و یا از header ی به نام x-access-token دریافت می کند و در صورتی که token ، valid بود ، decode شده آنرا به صورت یک property ، به req در middle ware اضافه می کنیم و چون بلافاصله بعد از آن وارد متد ‘/me’ می شویم ، این مق دار در req مربوط به این متد باقی مانده است .
    3. شماره 3) بعد از ریختن decoded در req در middle ware ف با صدا زدن متد next ، باعث می شویم که در واقع این متد شدا زده شود .
73. 
74. روش بعدی که استفاده از passport هست در برنامه ها نوشته شده .( برنامه هایی که با node نوشتیم و با social media های مختلف آنها را implement کردیم برای login .