

## Assignment-5

### Create an express API which verifies the Login

#### Procedure:

Create a folder ,install the express using cmd “`npm install express --save`”

also install the “CORS” “`npm install CORS --save`”

install mongodb “`npm i mongodb --save`”

now create a file api.js

#### api.js

```
const express=require('express');
const url=require('url');
const cors=require('cors');

const {MongoClient}=require('mongodb');
const mongourl='mongodb://127.0.0.1:27017';
const client=new MongoClient(mongourl);
const dbName='Ammu';
var api=express()
api.use(cors())
api.get('/',function(request,response){
    response.send("API server is online")
})
api.get('/insertnotes',async function(request,response){
    urldata=url.parse(request.url,true);
    var notes=urldata.query.notes;
    var owner=urldata.query.owner;
    console.log(notes,owner);
    await client.connect();
    console.log('connected with mongodb server');
    const db=client.db(dbName);
    const collection=db.collection('expressnotes');
    const result=await collection.insertOne({'notes':notes,'owner':owner});
    console.log(result);
    response.json({status:'success'});
})
api.get('/newregistration',async function(request,response){
    urldata=url.parse(request.url,true);
    var name=urldata.query.name;
    var mobile=urldata.query.mobile;
    var password=urldata.query.password;
    var email=urldata.query.email;
    var org=urldata.query.org;
    console.log(name,mobile,password,email,org);
    await client.connect();
```

```

    console.log('connected with mongodb server');
    const db=client.db(dbName);
    const collection=db.collection('expregister');
    const result=await
collection.insertOne({'name':name,'email':email,'mobile':mobile,'password':pas
sword,'org':org});
    console.log(result);
    response.json({status:'success'});
})
api.get('/verifylogin',async function(request,response){
    urldata=url.parse(request.url,true);
    var username=urldata.query.username;
    var password=urldata.query.password;
    console.log(username,password);
    await client.connect();
    console.log('connected to mongodb server');
    const db=client.db(dbName);
    const collection=db.collection('expregister');
    const result=await collection.find().toArray();
    console.log(result);
    var flag=0;
    for(let item of result){
        if(item['mobile']==username && item['password']==password){
            console.log('valid details')
            flag=1;}
    }
    if (flag==0)
    {
        response.json({status:'failure'});
    }
    else{
        response.json({status:'success'});
    }
})
api.listen(2000,function(){
    console.log('API server Online');
})

```

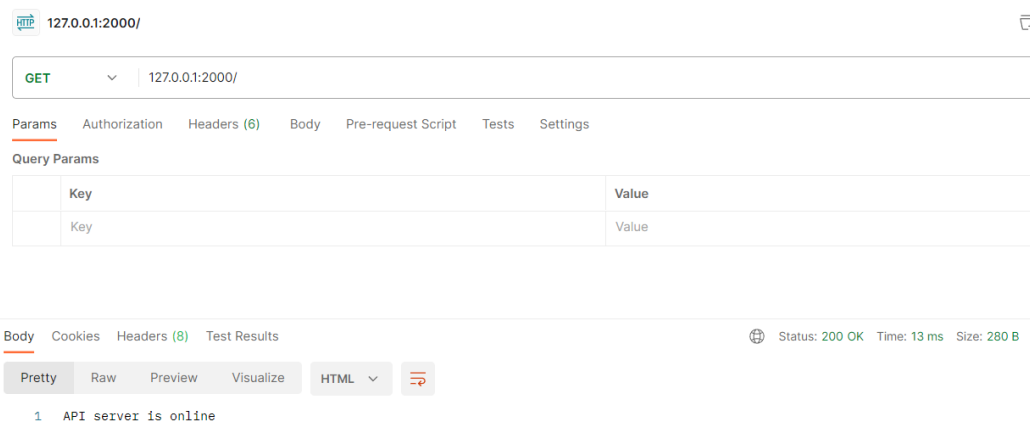
Run the code “ node api.js”

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

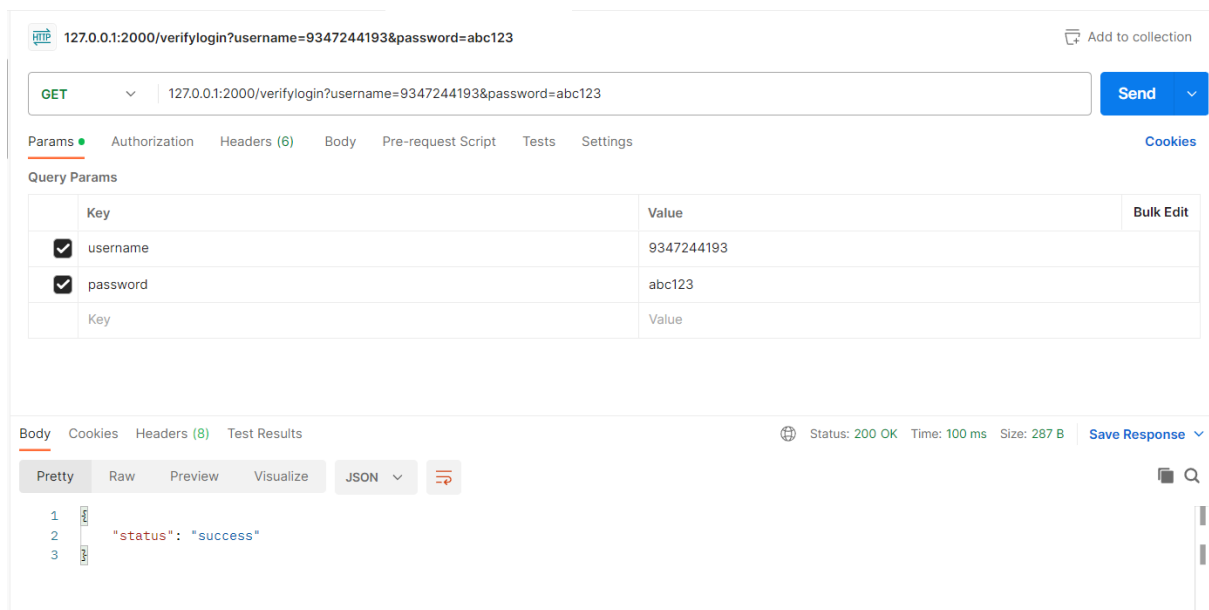
PS D:\s_internship\notes from url> node api.js
API server Online
█
```

Now ,at the server side,the api server is in online.

Open the postman tool app and select the get method,and call the default handler



The server side,also is on online,now call the “/verifylogin” handler and enter the values



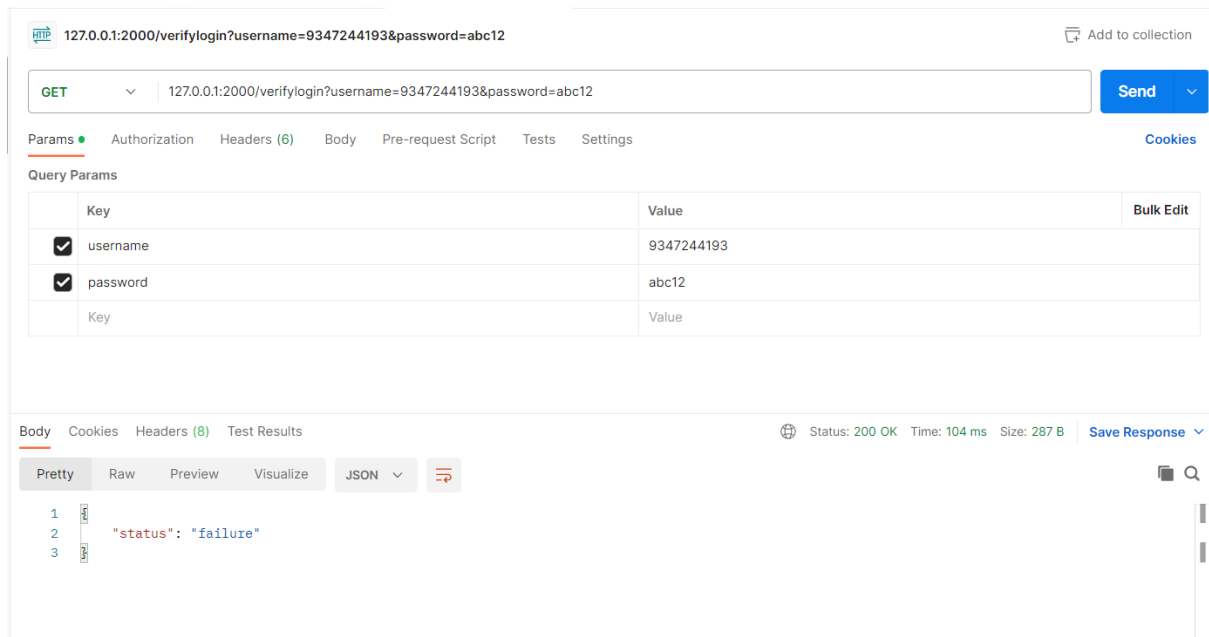
If the username,password is correct ,then it display the status : success

Now,the status is success. The inserted data is visible at the server side

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS D:\s_internship\notes from url> node api.js
API server Online
9347244193 abc123
connected to mongodb server
[
  {
    _id: new ObjectId("6485a90705f0126bab3647db"),
    name: 'Ammu',
    email: 'ammu123@gmail.com',
    mobile: '9347244193',
    password: 'abc123',
    org: 'makeskilled'
  }
]
valid details
[]
```

If the username,password is incorrect ,then it display the status : failure



The screenshot shows a REST client interface for a GET request to `127.0.0.1:2000/verifylogin?username=9347244193&password=abc12`. The request is successful (Status: 200 OK), but the response body indicates a failure.

Key	Value	Bulk Edit
<input checked="" type="checkbox"/> username	9347244193	
<input checked="" type="checkbox"/> password	abc12	
Key	Value	

Body: Cookies Headers (8) Test Results

Status: 200 OK Time: 104 ms Size: 287 B Save Response

```
1 {
2   "status": "failure"
3 }
```