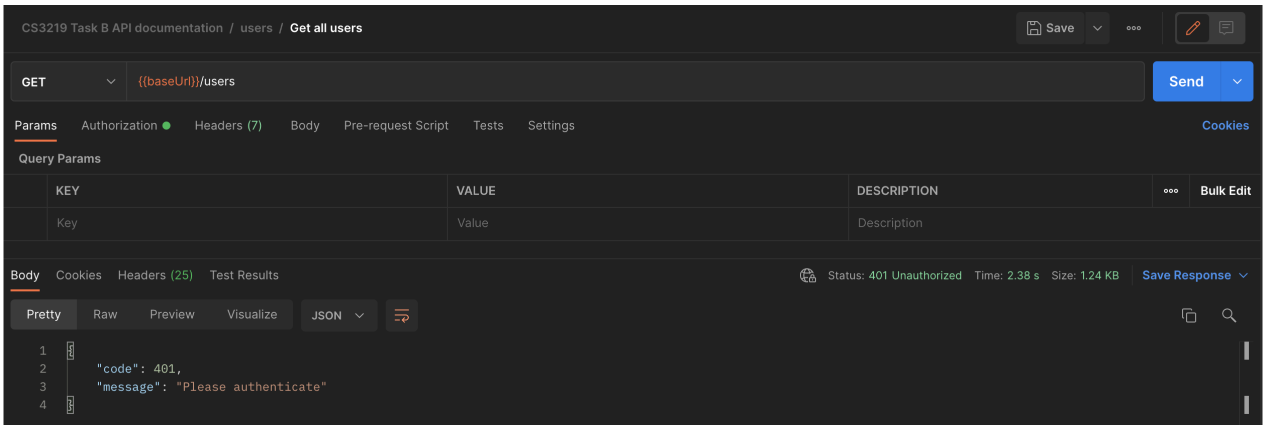
Submission Information

Option	Description
Name	Lau Jun Hao Benjamin
Matriculation Number	A01840840B
Link to GitHub Repository	(Same as task B) https://github.com/Capeguy/cs3219-otot-b
Instructions	Below
Other Relevant Learnings	null

Task C

Unsuccessful GET request to API Endpoint while Unauthenticated


CS3219 Task B API documentation / users / Get all users

GET {{baseUrl}}/users

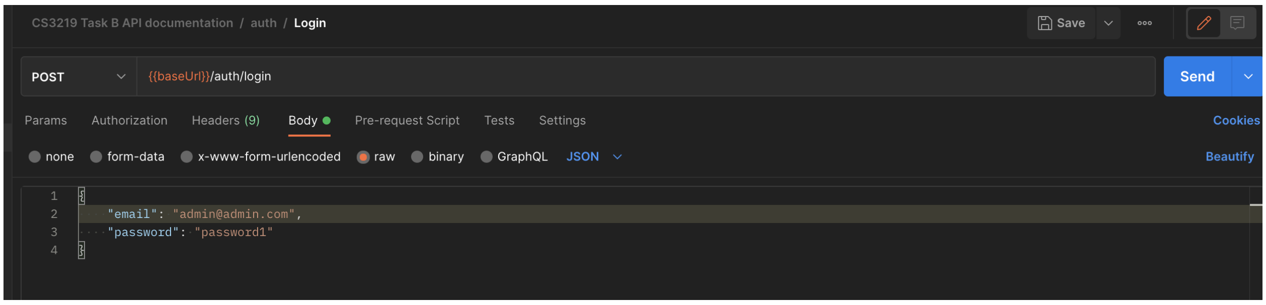
Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 401 Unauthorized Time: 2.38 s Size: 1.24 KB

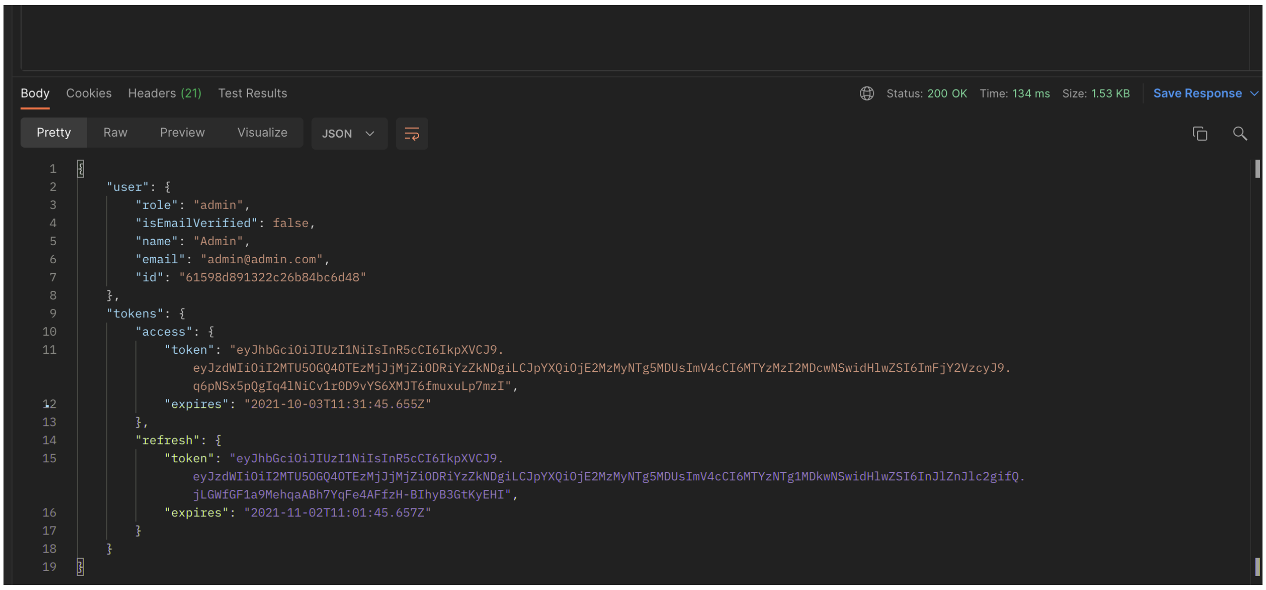
```
{
    "code": 401,
    "message": "Please authenticate"
}
```

Login as Admin (Full Permissions) / Authenticate User


CS3219 Task B API documentation / auth / Login

POST {{baseUrl}}/auth/login

```
{
    "email": "admin@admin.com",
    "password": "password1"
}
```

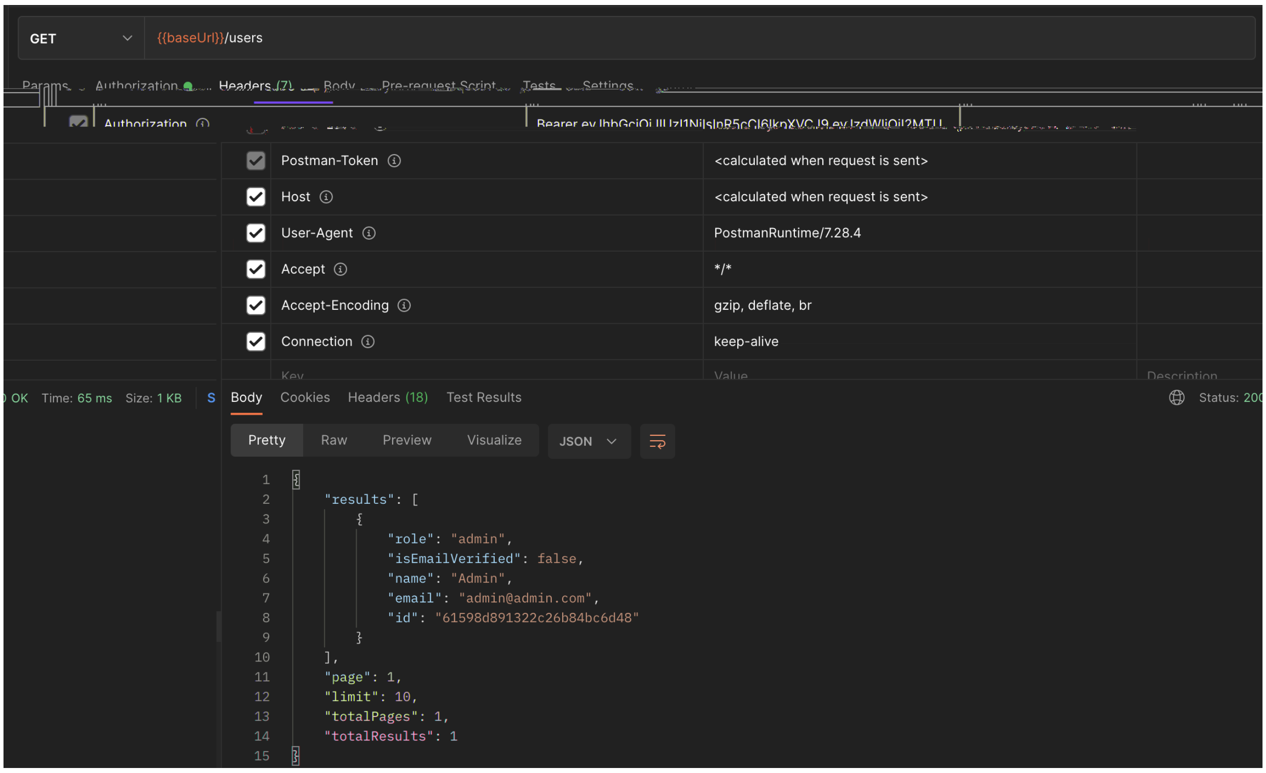


Body · Cookies · Headers (21) · Test Results · Status: 200 OK · Time: 134 ms · Size: 1.53 KB · Save Response

Pretty · Raw · Preview · Visualize · JSON

```
{
    "user": {
        "role": "admin",
        "isEmailVerified": false,
        "name": "Admin",
        "email": "admin@admin.com",
        "id": "61598d891322c26b84bc6d48"
    },
    "tokens": {
        "access": {
            "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
                eyJzdWIiOiI2MTU5OGQ4OTEzMjJjMjZiODRiYzZkNDgiLCJpYXQiOjE2MzMyNTg5MDUsImV4cCI6MTYzMzI2MDcwNSwidHlwZSI6ImFjY2VzcyJ9.
                q6pNSx5pQgIq4lNiCv1r0D9vYS6XMJT6fmuxuLp7mzI",
            "expires": "2021-10-03T11:31:45.655Z"
        },
        "refresh": {
            "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
                eyJzdWIiOiI2MTU5OGQ4OTEzMjJjMjZiODRiYzZkNDgiLCJpYXQiOjE2MzMyNTg5MDUsImV4cCI6MTYzNTg1MDkwNSwidHlwZSI6InJlZnJlc2gifQ.
                jLGWfGF1a9MehqaABh7YqFe4AFfzH-BIhyB3GtKyEHI",
            "expires": "2021-11-02T11:01:45.657Z"
        }
    }
}
```

Successful GET request to API Endpoint while Authenticated



GET {{baseUrl}}/users

Params · Authorization · Headers (7) · Body · Pre-request Script · Tests · Settings

	Key	Value	Description
✓	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI2MTU	
✓	Postman-Token	<calculated when request is sent>	
✓	Host	<calculated when request is sent>	
✓	User-Agent	PostmanRuntime/7.28.4	
✓	Accept	*/*	
✓	Accept-Encoding	gzip, deflate, br	
✓	Connection	keep-alive	

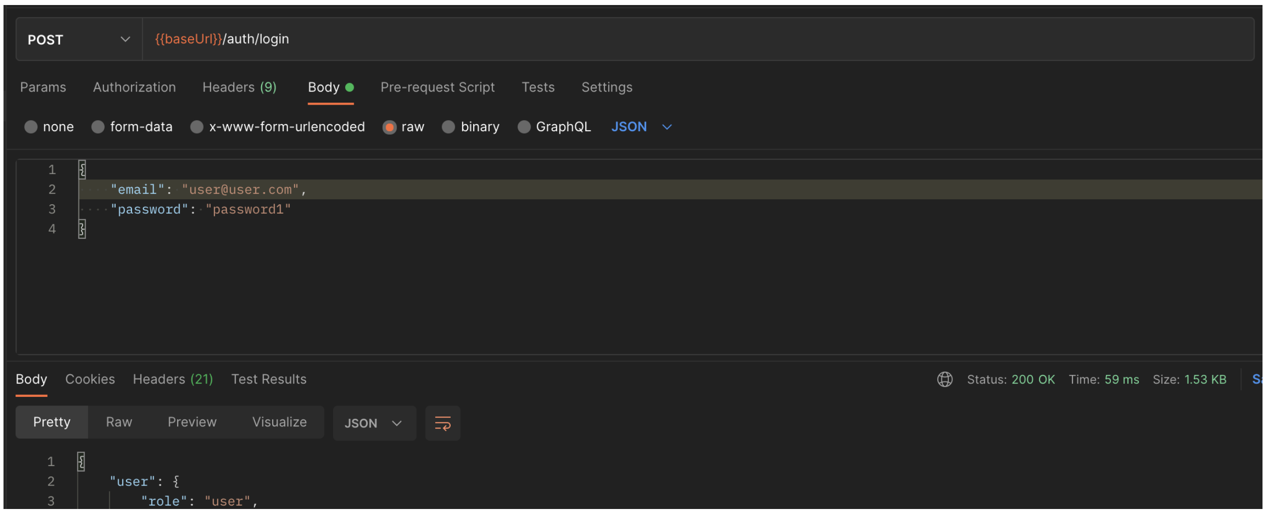
OK · Time: 65 ms · Size: 1 KB

Body · Cookies · Headers (18) · Test Results · Status: 200

Pretty · Raw · Preview · Visualize · JSON

```
{
    "results": [
        {
            "role": "admin",
            "isEmailVerified": false,
            "name": "Admin",
            "email": "admin@admin.com",
            "id": "61598d891322c26b84bc6d48"
        }
    ],
    "page": 1,
    "limit": 10,
    "totalPages": 1,
    "totalResults": 1
}
```

Login as User (Limited Permissions) / Authenticate User



POST {{baseUrl}}/auth/login

Params · Authorization · Headers (9) · Body ● · Pre-request Script · Tests · Settings

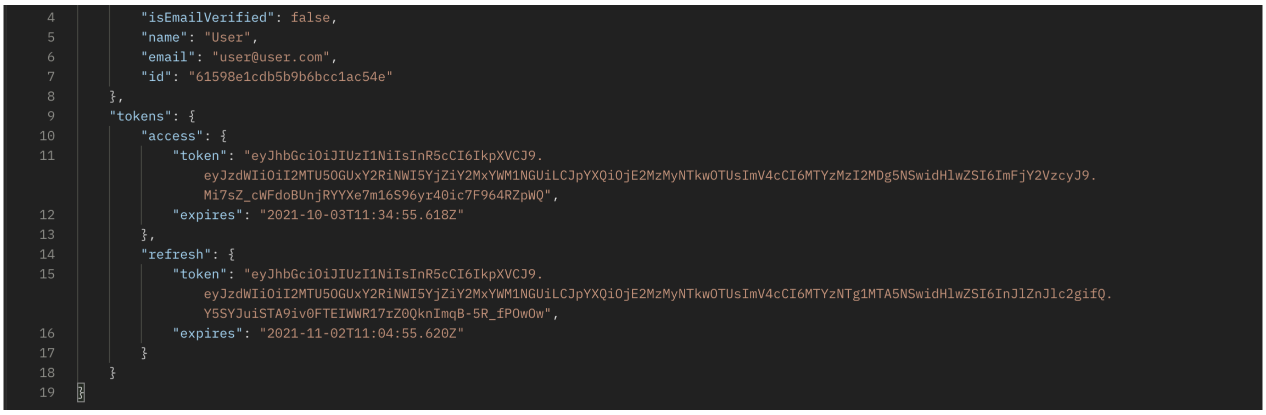
none · form-data · x-www-form-urlencoded · [✓ raw] · binary · GraphQL · JSON

```
{
    "email": "user@user.com",
    "password": "password1"
}
```

Body · Cookies · Headers (21) · Test Results · Status: 200 OK · Time: 59 ms · Size: 1.53 KB

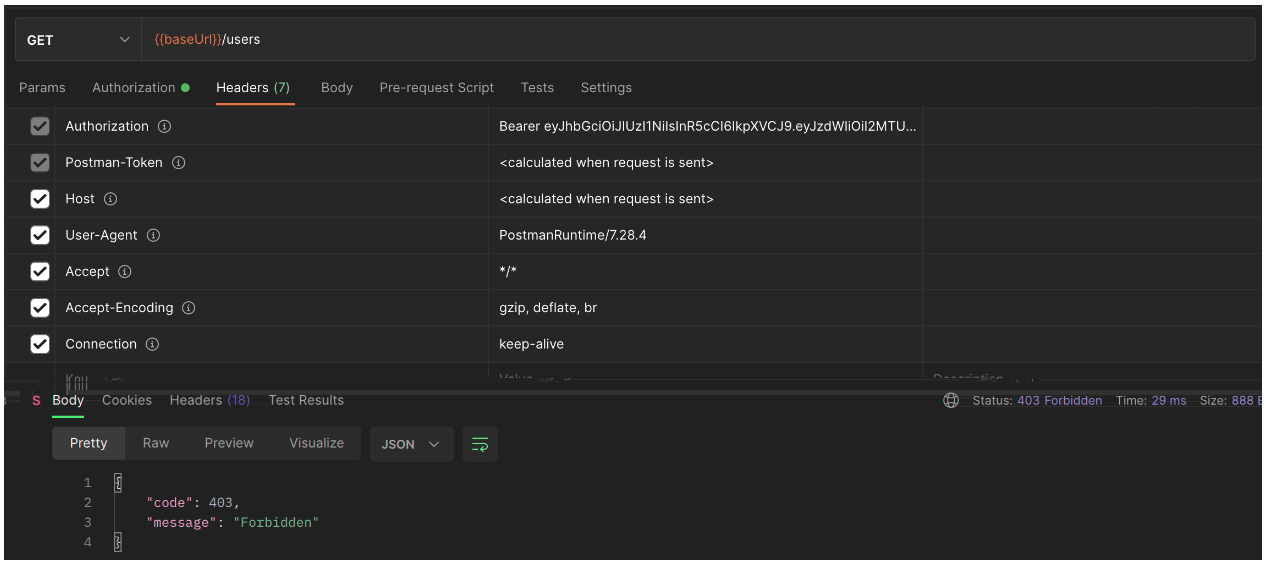
Pretty · Raw · Preview · Visualize · JSON

```
{
    "user": {
        "role": "user",
```



```
        "isEmailVerified": false,
        "name": "User",
        "email": "user@user.com",
        "id": "61598e1cdb5b9b6bcc1ac54e"
    },
    "tokens": {
        "access": {
            "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
                eyJzdWIiOiI2MTU5OGUxY2RiNWI5YjZiY2MxYWM1NGUiLCJpYXQiOjE2MzMyNTkwOTUsImV4cCI6MTYzMzI2MDg5NSwidHlwZSI6ImFjY2VzcyJ9.
                Mi7sZ_cWFdoBUnjRYYXe7m16S96yr40ic7F964RZpWQ",
            "expires": "2021-10-03T11:34:55.618Z"
        },
        "refresh": {
            "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
                eyJzdWIiOiI2MTU5OGUxY2RiNWI5YjZiY2MxYWM1NGUiLCJpYXQiOjE2MzMyNTkwOTUsImV4cCI6MTYzNTg1MTA5NSwidHlwZSI6InJlZnJlc2gifQ.
                Y5SYJuiSTA9iv0FTEIWWR17rZ0QknImqB-5R_fPOwOw",
            "expires": "2021-11-02T11:04:55.620Z"
        }
    }
}
```

Unsuccessful GET request to API Endpoint while Unauthorized



GET {{baseUrl}}/users

Params | Authorization ● | Headers (7) | Body | Pre-request Script | Tests | Settings

✓	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI2MTU...
✓	Postman-Token	<calculated when request is sent>
✓	Host	<calculated when request is sent>
✓	User-Agent	PostmanRuntime/7.28.4
✓	Accept	*/*
✓	Accept-Encoding	gzip, deflate, br
✓	Connection	keep-alive

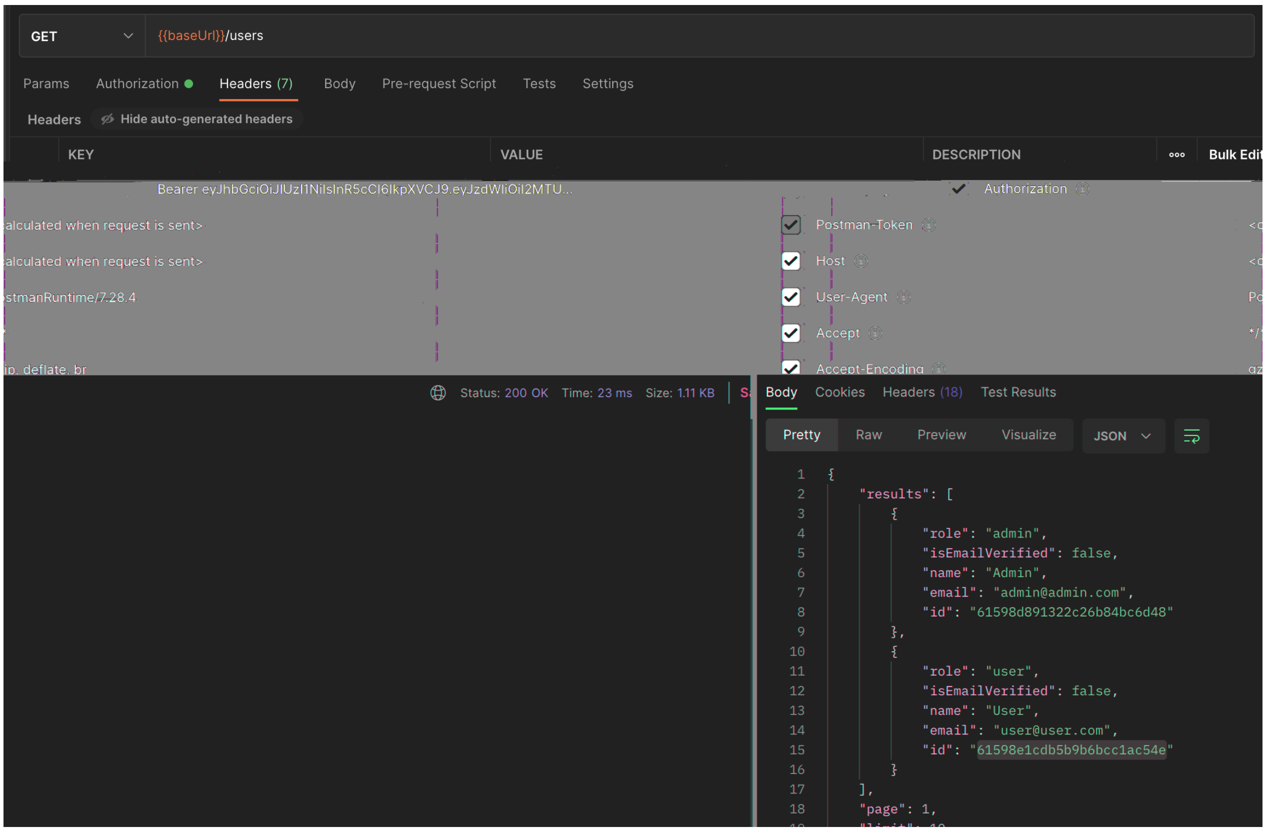
Body | Cookies | Headers (18) | Test Results — Status: 403 Forbidden Time: 29 ms Size: 888 B

Pretty | Raw | Preview | Visualize | JSON

```
{
    "code": 403,
    "message": "Forbidden"
}
```

After granting the User rights:

Successful GET request to API Endpoint while Unauthorized



GET {{baseUrl}}/users

Params | Authorization ● | Headers (7) | Body | Pre-request Script | Tests | Settings

Headers — Hide auto-generated headers

	KEY	VALUE	DESCRIPTION
✓	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI2MTU...	
✓	Postman-Token	<calculated when request is sent>	
✓	Host	<calculated when request is sent>	
✓	User-Agent	PostmanRuntime/7.28.4	
✓	Accept	*/*	
✓	Accept-Encoding	gzip, deflate, br	

Status: 200 OK Time: 23 ms Size: 1.11 KB

Body | Cookies | Headers (18) | Test Results

Pretty | Raw | Preview | Visualize | JSON

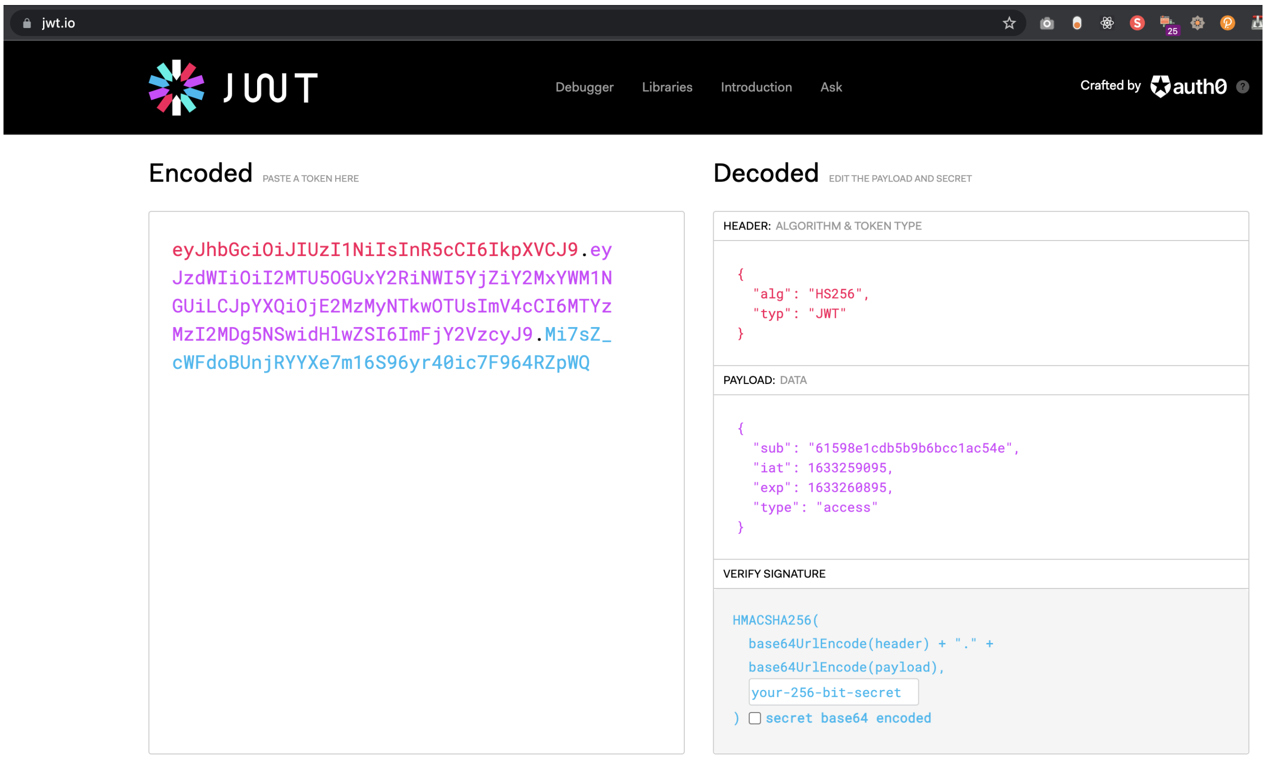
```
{
    "results": [
        {
            "role": "admin",
            "isEmailVerified": false,
            "name": "Admin",
            "email": "admin@admin.com",
            "id": "61598d891322c26b84bc6d48"
        },
        {
            "role": "user",
            "isEmailVerified": false,
            "name": "User",
            "email": "user@user.com",
            "id": "61598e1cdb5b9b6bcc1ac54e"
        }
    ],
    "page": 1,
    "limit": 10,
    "totalPages": 1,
    "totalResults": 2
}
```

Implementation

Usage of JWT

Calling the /login endpoint successfully will return Access and Refresh Tokens.

Breakdown of Access Token



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiI2MTU5OGUxY2RiNWI5YjZiY2MxYWM1N
GUiLCJpYXQiOjE2MzMyNTkwOTUsImV4cCI6MTYz
MzI2MDg5NSwidHlwZSI6ImFjY2VzcyJ9.Mi7sZ_
cWFdoBUnjRYYXe7m16S96yr40ic7F964RZpWQ
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

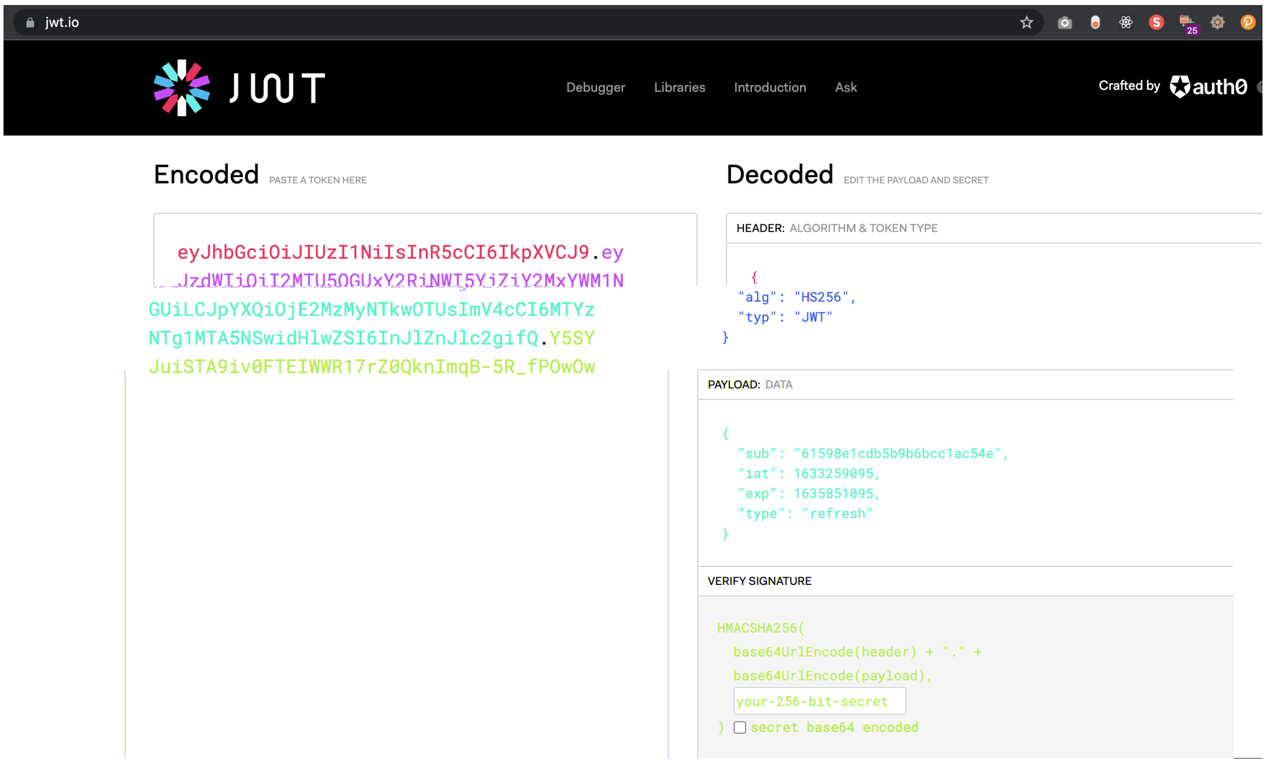
PAYLOAD: DATA

```
{
  "sub": "61598e1cdb5b9b6bcc1ac54e",
  "iat": 1633259095,
  "exp": 1633260895,
  "type": "access"
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

Breakdown of Refresh Token



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiI2MTU5OGUxY2RiNWI5YjZiY2MxYWM1N
GUiLCJpYXQiOjE2MzMyNTkwOTUsImV4cCI6MTYz
NTg1MTA5NSwidHlwZSI6InJlZnJlc2gifQ.Y5SY
JuiSTA9iv0FTEIWWR17rZ0QknImqB-5R_fPOwOw
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "61598e1cdb5b9b6bcc1ac54e",
  "iat": 1633259095,
  "exp": 1635851095,
  "type": "refresh"
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

In order to use all other APIs which require authentication, the Authentication Bearer header must be included with the Access Token, otherwise a 401 Unauthorised will be returned.

with the Access Token, otherwise a 401 Unauthorised will be returned.

Use of Framework with Role and Permissions Support

By using the passport.js authentication middleware we are able to define what rights are required for each API endpoint.

```
const express = require('express');
const auth = require('../../middlewares/auth');
const validate = require('../../middlewares/validate');
const userValidation = require('../../validations/user.validation');
const userController = require('../../controllers/user.controller');

const router = express.Router();

router
  .route('/')
  .post(auth('manageUsers'), validate(userValidation.createUser), userController.createUser)
  .get(auth('getUsers'), validate(userValidation.getUsers), userController.getUsers);

router
  .route('/:userId')
  .get(auth('getUsers'), validate(userValidation.getUser), userController.getUser)
  .patch(auth('manageUsers'), validate(userValidation.updateUser), userController.updateUser)
  .delete(auth('manageUsers'), validate(userValidation.deleteUser), userController.deleteUser);

module.exports = router;
```

In the above example, the `POST` to `/` requires the `manageUsers` right, which ensures that the authenticated user has that right before proceeding.

The Role to Rights mapping is maintained in `src/config/roles.js` and easily extensible to define more roles and rights.

```
const allRoles = {
  user: [],
  admin: ['getUsers', 'manageUsers'],
};

const roles = Object.keys(allRoles);
const roleRights = new Map(Object.entries(allRoles));

module.exports = {
  roles,
  roleRights,
};
```