

Testspec krav1.3

Last edited by [Julia Lind](#) 1 week ago

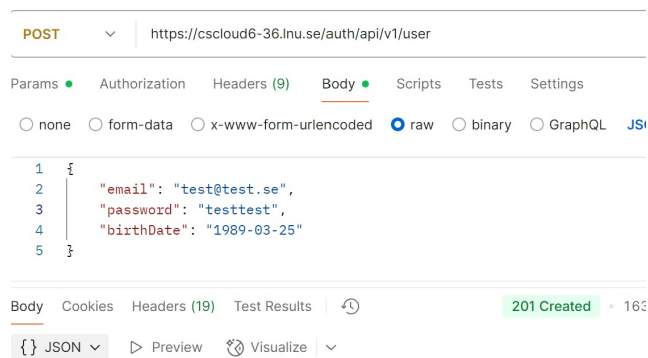
1.3 Tokenhantering

Pre-condition

1. Skapa en användare via postman genom att göra en POST request till <https://cscloud6-36.lnu.se/auth/api/v1/user> och skicka följande objekt:

```
{  
  "email": "test@test.se",  
  "password": "testtest",  
  "birthDate": "1989-03-25"  
}
```

Om allt gick bra ska du få status 201 tillbaka.



2. Gör POST request till <https://cscloud6-36.lnu.se/auth/api/v1/login> med följande objekt

```
{  
  "email": "test@test.se",  
  "password": "testtest"  
}
```

Om allt gick bra ska du få tillbaka två tokens:

1. Access token ska gälla två timmar

Action

Kopiera mittendelen av accessToken (dvs det som är mellan två punkter, punkterna ej inkluderat).

[illegible]

Navigera till <https://www.base64decode.org/> i web-browsern och klistra in den kopierade strängen i fältet "Decode from Base64 format". Klicka sedan på "Decode" knappen.

Kopiera siffran från "exp" fältet och navigera till <https://www.unixtimestamp.com/>, klistra in siffran i fältet "Enter a timestamp" och klicka på "Convert" knappen.

The Current Epoch Unix Timestamp

Enter a Timestamp

1747700001

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Convert →

1747680000

SECONDS SINCE 12:00:00 AM, JAN 1, 1970

| | |
|--------|-----------------------------------|
| Format | Seconds |
| GMT | Tue May 20 2025 00:13:21 GMT+0000 |

Gör samma sak med siffran i "iat" fältet:

Enter a Timestamp

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

[Convert →](#)

1747

SECONDS SIN

12

[Copy](#)

| | |
|--------|-----------------------------------|
| Format | Seconds |
| GMT | Mon May 19 2025 22:13:21 GMT+0000 |

Expected outcome

Det ska vara exakt två h mellan de båda tiderna:

2. Refresh token ska gälla två dygn

Action

Kopiera mittendelen av refresh-tokenet som du fått från servern (dvs det som är emellan de två punkterna, punkterna ej inräknat)

```
1IKIDGa6C8KdTKZ8D5W4YcQUIZ16OK_41La6Zs1IdB9n0Z1/1MXGR1dV5YX_K1Du0Nna091Zm1MKjnsWg",  
"refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJqdGkiOiI2ODJiYWQwMTEwOWI5ZmIyYzFkNDRIInZAIjE6IjE3NDc2OTI4MDEsImV4cCI6MTc0Nzg2NTYwMX0.  
dKsQZT3x8xvgY8vXHLMOtNGMC65pVVqZid9mvs0T_Es"
```

Navigera till <https://www.base64decode.org/> i web-browsern och klistra in den kopierade strängen i fältet "Decode from Base64 format". Klicka sedan på "Decode" knappen.

Decode from Base64 format

Simply enter your data then push the decode button.

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI2ODJiYWQwMTEwOWI5ZmIyYzFkNDRIInZAIjE6IjE3NDc2OTI4MDEsImV4cCI6MTc0Nzg2NTYwMX0

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8

Source character set.

☐

Decode each line separately (useful for when you have multiple entries).

☒

Live mode OFF

Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE >

Decodes your data into the area below.

{ "alg": "HS256", "typ": "JWT" }, { "jti": "682bad01109b9fb2c1d44b70", "iat": 1747692801, "exp": 1747865601 }

Kopiera siffran i fältet "exp". Navigera till <https://www.unixtimestamp.com/> och klistra in siffran i fältet "Enter a Timestamp" samt klicka på "Convert" knappen.

The Current Epoch Unix Timestamp

Enter a Timestamp

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Convert →

17476941

SECONDS SINCE JAN 01 1970. (UTC)

12:36:41 AM

Copy

| | |
|--------|-----------------------------------|
| Format | Seconds |
| GMT | Wed May 21 2025 22:13:21 GMT+0000 |

Gör samma sak för siffran i fältet "iat":

Enter a Timestamp

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Convert →

1747

SECONDS SINCE JAN 01 1970. (UTC)

Copy

| | |
|--------|-----------------------------------|
| Format | Seconds |
| GMT | Mon May 19 2025 22:13:21 GMT+0000 |

Expected outcome

Skillnaden mellan de båda tiderna ska vara exakt 48 h.

3. Återanvändning av refresh-token är inte tillåten och tokenrotation samt token chaining ska användas som säkerhetsmekanismer

Del 1 - OK

Action

1. i en ny Postman flik gör en POST request till <https://cscloud6-36.lnu.se/auth/api/v1/login> och skicka med följande objekt:

```
{  
  "email": "test@test.se",  
  "password": "testtest"  
}
```

The screenshot shows the Swagger UI for a POST endpoint. The 'Body' tab is selected, showing the request body as a JSON object with 'email' and 'password' fields. The response body is also shown as a JSON object with 'accessToken' and 'refreshToken' fields. The status bar indicates a 201 Created response.

2. Kopiera därefter refresh.tokenet (dubbelfnuttar ej inkluderat). Öppna en ny flik i Postman, klicka på Authorization, välj "Bearer Token" under "Auth type" och klustra in tokenet som du kopierat i fliken "Token". Gör en Post request till <https://cscloud6-36.lnu.se/auth/api/v1/refresh>.

https://cscloud6-36.lnu.se/auth/api/v1/refresh

Save

POST

https://cscloud6-36.lnu.se/auth/api/v1/refresh

Send

Params

Authorization

Headers (9)

Body

Scripts

Tests

Settings

Auth Type

Bearer Token

Token

E3Ao8XbudWsd280VzRPmTrCw481E

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body

Cookies

Headers (19)

Test Results

201 Created

52 ms

1.57 KB

{ } JSON

Preview

Visualize

```
1 {
2   "accessToken": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIj7Im1kIjo1NjgyYmFiOTYxMDliOWZiMmMxZDQ0YjZkIiwiaWd1IjozNn0sIm1hdCI6MTMt0NzY5NjY2OSwiZXhwIjozQ3NzAzODY5fQ.QgnbpmLaiWfmxnd0ckr8Tm8ESpNKmn14iQnLgx3Ne2Vq10HWWVEQbZesaSE1LTYG6Sxm3C-1acxvnB8DsgltB3DWIPngacDb3Q6FA1oXRN7bk8p0Egb-KLGBF8CSrayx789uCXU7z81NFvy_DjY692zS1fIoZ5U7f03IZ0XEQRW6PytwUqt2d29PcQUzVP1N1W4GGSMIuDs5N6Bj5yZwKIbXa90FBo_rjsRlzfS8IqaF1tSwhyRh0gjxCesdqdw h8Pg4uZG5mQ6oPBtpvqazj9_Zr4nEE-tr1cknTG71Ub0a15MdHYAeo9N2M2F0pv8vujs7zuVtJ06Fqg",
3   "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI2ODJiYmMxZDEwIWI5ZmIyYzFkNDRIInR5cCI6IkpXVCJ9.eyJ1c2VyIj7Im1kIjo1NjgyYmFiOTYxMDliOWZiMmMxZDQ0YjZkIiwiaWd1IjozNn0sIm1hdCI6MTMt0NzY5NjY2OSwiZXhwIjozQ3NzAzODY5fQ.1t_1DGUxmwN8B5-Hz9rk9h-8Ag7MepSELxgeaH4Gww"
4 }
```

3. Kopiera refresh token från senaste responsen och upprepa steg 2
4. Anslut till MongoDB databasen för Authserver på csccloud via MondoDB Compass:

```
ssh -L 27019:localhost:27019 ubuntu@194.47.176.36
```

Expected outcome

I steg 1 ska du ha fått en ny access token och en ny refresh token som skiljer sig från de du fick i Pre-condition steget.

I de bägge efterföljande requesten till refreshrouten ska du ha fått ytterligare nya accesstokens och refreshtokens som inte är samma som tidigare.

De två refresh-tokens som du använt för att generera nya ska ha ett objektid i fältet för "next" samt ha "expired" attributet satt till "true"

```
_id: ObjectId('682bad01109b9fb2c1d44b70')
next: null
expired: false
user: ObjectId('682bab96109b9fb2c1d44b6d')
createdAt: 2025-05-19T22:13:21.170+00:00
updatedAt: 2025-05-19T22:13:21.170+00:00
```

```
_id: ObjectId('682bbbfa109b9fb2c1d44b73')
next: ObjectId('682bbc1d109b9fb2c1d44b77')
expired: true
user: ObjectId('682bab96109b9fb2c1d44b6d')
createdAt: 2025-05-19T23:17:14.604+00:00
updatedAt: 2025-05-19T23:17:49.043+00:00
```

```
_id: ObjectId('682bbc1d109b9fb2c1d44b77')
next: ObjectId('682bbc9d109b9fb2c1d44b7c')
expired: true
user: ObjectId('682bab96109b9fb2c1d44b6d')
createdAt: 2025-05-19T23:17:49.040+00:00
updatedAt: 2025-05-19T23:19:57.159+00:00
```

```
_id: ObjectId('682bbc9d109b9fb2c1d44b7c')
next: null
expired: false
user: ObjectId('682bab96109b9fb2c1d44b6d')
createdAt: 2025-05-19T23:19:57.156+00:00
updatedAt: 2025-05-19T23:19:57.156+00:00
```

Del 2 - Not OK

Action

1. i en ny Postman flik gör en POST request till <https://cscloud6-36.lnu.se/auth/api/v1/login> och skicka med följande objekt:

```
{
  "email": "test@test.se",
  "password": "testtest"
}
```

POST <https://cscloud6-36.lnu.se/auth/api/v1/login>

Params • Authorization Headers (9) **Body** • Scripts Tests Settings

☐ none ☐ form-data ☐ www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▾

```

1 {
2   "email": "test@test.se",
3   "password": "testtest"
4 }

```

Body Cookies Headers (19) Test Results ⌚

{ } JSON ▾ ▶ Preview 🔗 Visualize ▾

```

1 {
2   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyYjP7ImNyZWZlbnRlcCI6IiwmbmVudCMTUMjIjEwMDc0MTRgLnjiIWIiOiJmVWZGF0ZWRRBdC.iOTYxMDliOWZiMmMxZDQ0YjZkIiwiaWF0IjoxNnBsImhhdCI6MTc0NzY5Njk2NCwiZXhwIjoxNzQ3NWVqcyYxfgXcKNoaClucix-j3h8tvqSct2Nqla1e1cLSkCWie69BhcduDYZTQfkerUGYvPZTI0BhdJFRFBIABW_xittkCwiy7PUyaQP1rB0i8RAwa8c5ZqKoTrZvVfjPmMUODITTTuFk_qW53lN9PxskYKLvwlbXVH2U3pw6qOigZ-iI_nnnw8jyfPT2DasBVirWv4fupuz-MCG3DA8puIccz7tynuMLr-m2IoFXqdeP"
3   "refreshToken": "eyJ3bGciOiJ1UzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJ1ODI0ZjYmQDEWEWISZmIyZyZkNDRI0DAiLCJpYXQiOiE3NDc2OTY5NjQsImV4cCI6MTc0GIzdnhK0BGXBBI2ohaD-93znZD22m2whJ_b0vTw"
4 }

```

2. Kopiera refresh tokenet (dubbelnuttar ej inkluderat). Öppna en ny flik i Postman, klicka på Authorization, välj "Bearer Token" under "Auth type" och klistra in tokenet som du kopierat i fliken "Token". Gör en Post request till <https://cscloud6-36.lnu.se/auth/api/v1/refresh>.

POST <https://cscloud6-36.lnu.se/api/v1/refresh>

Params **Authorization** Headers (9) Body Scripts Tests Settings

Auth Type

Bearer Token

TOKEN

Biz0haD-93zNzZD2D2mZh_Lb0Vtw

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body Cookies Headers (19) Test Results 201 Created

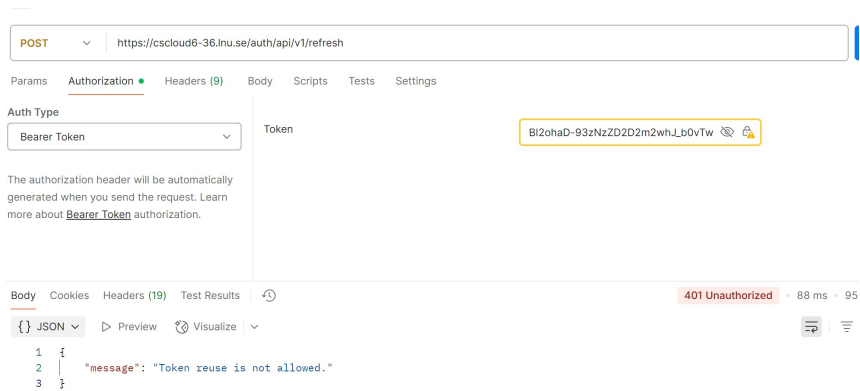
```
{ "JSON": {
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjpbImkiOiwiYmF5bmFiOTYxMDIwZWZlMmMzMjQ0QyZkIiwiaWF0IjzoInNoeSImIhdC16MTc0NzY5NmZuA5NyYmIjozeXNzQ3NmA0Mjg5eX5fWGBRDG9C8vjiM3JlVdNn-f6kYhw4pskInXpCr--XUKHjQdZK9No4dQrLG03rtA3ojFGwV0TG-97KQQYVQp2y1q7058vUR3Du0C2KjptbxbX83VveTbWkk4o78PTmjCz5AzCe7-gtozz1E0x25B7kpflo5J38d2msxYXdGSD3n3VuGBBzanddd0j6E6ZH-cuARqzaPfb2hAuxUX3CMoLIqASkL6i2AU3ynin3Ho7qKUAuz_e33yZfXuieFNak_6e6WzFKQG_tETs2CiYXLedRLT2-pwjRAN8rfvnjYmwv",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI2ODIiYmRjOTcwOWI5ZmZyZfKNDRI0DQilc2JpYXQiOjE3NDI0C2t0CwTotsImV4cCI6MTc0Nzg2OTgtg5N30.WxlWh_EvakYkAsHmgQ8P4skFswN24DNbhLlp1leaPU"
```

4. Kopiera och spara refresh token som du har fått till senare. Klicka på send-knappen igen så att du gör en ny förfrågan till refresh routen med samma token.
5. Använd nu den refresh token som du fick i steg 3 och gör en ny förfrågan till refresh routen.
6. Anslut till MongoDB databasen för Authserver på csccloud via MondoDB Compass:

```
ssh -L 27019:localhost:27019 ubuntu@194.47.176.36
```

Expected outcome

I både steg 5 och 6 ska du få 401 tillbaka. Eftersom det första tokenet återanvändes så expirerades hela kedjan som skapats ur den.



POST <https://csccloud6-36.lnu.se/auth/api/v1/refresh>

Params Authorization Headers (9) Body Scripts Tests Settings

Auth Type: Bearer Token

Token: BI2ohaD-93zNzZD2D2m2wh_Lb0vTw

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Body Cookies Headers (19) Test Results

401 Unauthorized - 88 ms - 95

```
{
  "message": "Token reuse is not allowed."
}
```

I MongoDB compass ser du att det senaste tokenet har blivit markerat som "Expired" men inte ersatts med någon ny (Next)

```
{
  "_id": ObjectId('682bbd44109b9fb2c1d44b80'),
  "next": ObjectId('682bbdc9109b9fb2c1d44b84'),
  "expired": true,
  "user": ObjectId('682bab96109b9fb2c1d44b6d'),
  "createdAt": 2025-05-19T23:22:44.491+00:00,
  "updatedAt": 2025-05-19T23:24:57.928+00:00
}
```

```
{
  "_id": ObjectId('682bbdc9109b9fb2c1d44b84'),
  "next": null,
  "expired": true,
  "user": ObjectId('682bab96109b9fb2c1d44b6d'),
  "createdAt": 2025-05-19T23:24:57.926+00:00,
  "updatedAt": 2025-05-19T23:26:12.407+00:00
}
```

Cleanup

Ta bort användaren genom att skicka en DELETE request till <https://csccloud6-36.lnu.se/auth/api/v1/user> med följande objekt:

```
{
  "email": "test@test.se",
  "password": "testtest"
}
```