

W13-P1: investigate jwt token using jwt.io

The screenshot shows the jwt.io interface with a valid JWT token entered into the text area. The token is:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJzdXBhYmFzZSIsInJlZii6Imh6bGx3a2l4aWp1b3Ficm9wbmF0Iiwicm9sZSI6ImFub24iLCJpYXQiOjE2NzY0NTg2ODYsImV4cI6MTk5MjAzNDY4Nn0.jnQZ1xX-Iem458H00h05nj-OjDwAX11Q21HtJ6HoWPs
```

The token is highlighted with a red border. The interface shows the decoded header, payload, and verify signature sections. The payload section contains fields like "iss", "ref", "role", "iat", and "exp". A "Signature Verified" message is displayed in a box at the bottom left, and a "SHARE JWT" button is at the bottom right.

<https://jwt.io>

The screenshot shows the jwt.io interface with an encoded JWT token entered into the "Encoded" text area. The token is:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJldmVyeWRzMrmZGdkZmdkZiIsInJlZi6InNlcnPzYVfcn9sZSIsInJvbGUiOjJhbm9uIiwiWF0IjoxNjc2NDU4Njg2LCJleHA1ojE5OTIwMzQ2ODZ9.AdIdUz1P3Ijxx87eFZSm8CzTP1tikBSvzhak3uSzfOQ
```

The token is highlighted with a red border. The interface shows the "Decoded" section where the token is pasted, revealing the same header and payload as above. The payload section contains fields like "iss", "ref", "role", "iat", and "exp". A "Signature Verified" message is displayed in a box at the bottom left, and a "SHARE JWT" button is at the bottom right.

W13-P2: Get all menu data from menu_xx (no RLS), and from menu2_xx (RLS enabled)

Get all menu from menu_xx (RLS not enabled)

Postman Request:

```
GET https://hzllwkixijuqbrotnat.supabase.co/rest/v1/menu_96?select=*
Headers:
  Connection: keep-alive
  apikey: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJzdXBhYmFzZSIinJlZlI6Inmh6bGx3a2l4aWp1b3Ficm9wbnF0Iiwicm9sZSI6ImFub24iLCJpYXQiOjE2NzY0NTg2ODYsImV4cCI6MTk5MjAzNDY4Nm0.gASAx7BiiJEzsSKLct2WjivgUhzbFD1CjJSU08k2WY"
  Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJzdXBhYmFzZSIinJlZlI6Inmh6bGx3a2l4aWp1b3Ficm9wbnF0Iiwicm9sZSI6ImFub24iLCJpYXQiOjE2NzY0NTg2ODYsImV4cCI6MTk5MjAzNDY4Nm0.gASAx7BiiJEzsSKLct2WjivgUhzbFD1CjJSU08k2WY"
```

Supabase API Docs - menu_96:

To read rows in menu_96, use the select method.

curl

```
'https://hzllwkixijuqbrotnat.supabase.co/rest/v1/menu_96?select='`
```

curl

```
-H "apikey: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJzdXBhYmFzZSIinJlZlI6Inmh6bGx3a2l4aWp1b3Ficm9wbnF0Iiwicm9sZSI6ImFub24iLCJpYXQiOjE2NzY0NTg2ODYsImV4cCI6MTk5MjAzNDY4Nm0.gASAx7BiiJEzsSKLct2WjivgUhzbFD1CjJSU08k2WY"
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3Mi0iJzdXBhYmFzZSIinJlZlI6Inmh6bGx3a2l4aWp1b3Ficm9wbnF0Iiwicm9sZSI6ImFub24iLCJpYXQiOjE2NzY0NTg2ODYsImV4cCI6MTk5MjAzNDY4Nm0.gASAx7BiiJEzsSKLct2WjivgUhzbFD1CjJSU08k2WY"
```

curl

```
'https://hzllwkixijuqbrotnat.supabase.co/rest/v1/menu_96?select=some_column,other_col'
```

Fail to get all menu from menu2_xx (RLS enabled)

The screenshot displays two windows: Postman on the left and the Supabase web interface on the right.

Postman: A GET request is made to `https://hzllwkixijuqbropnat.supabase.co/rest/v1/menu2_96?select=*`. The Headers tab shows the following configuration:

- Accept: */*
- Accept-Encoding: gzip, deflate, br
- Connection: keep-alive
- apikey: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ...
- Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJp...

The response status is 200 OK, and the time taken is 1457 ms.

Supabase (Auth | Supabase): The 'menu2_96' table has RLS enabled. The Policies section shows the message: "No policies created yet". It also includes a link to "Create policy" and a "Disable RLS" button.

to get all menu from menu2_xx (RLS enabled) by providing service_role token

Get all menu data from menu_xx (no RLS), and from menu2_xx (RLS enabled)

W13-P3: Get all category=breakfast data from menu_xx (no RLS), and from menu2_xx (RLS enabled)

Get all category=breakfast data from menu_xx (RLS not enabled)

The screenshot shows the Postman interface with a successful API call. The URL is `https://hzilwixijjuoqbropnat.supabase.co/rest/v1/menu_96?category=eq.breakfast&select=*`. The response body is a JSON array containing two items, both of which are highlighted with red boxes.

```
8 |     "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam."
9 |
10 | },
11 | {
12 |     "id": 5,
13 |     "title": "Egg Attack",
14 |     "category": "breakfast",
15 |     "price": 2.99,
16 |     "img": "/menu/images/item-5.jpeg",
17 |     "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
18 | },
19 | {
20 |     "id": 7,
21 |     "title": "Bacon Overflow",
22 |     "category": "breakfast",
23 |     "price": 8.99,
24 |     "img": "/menu/images/item-7.jpeg",
25 |     "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
```

get all category=breakfast data from menu2_xx (RLS enabled) by providing service_role token

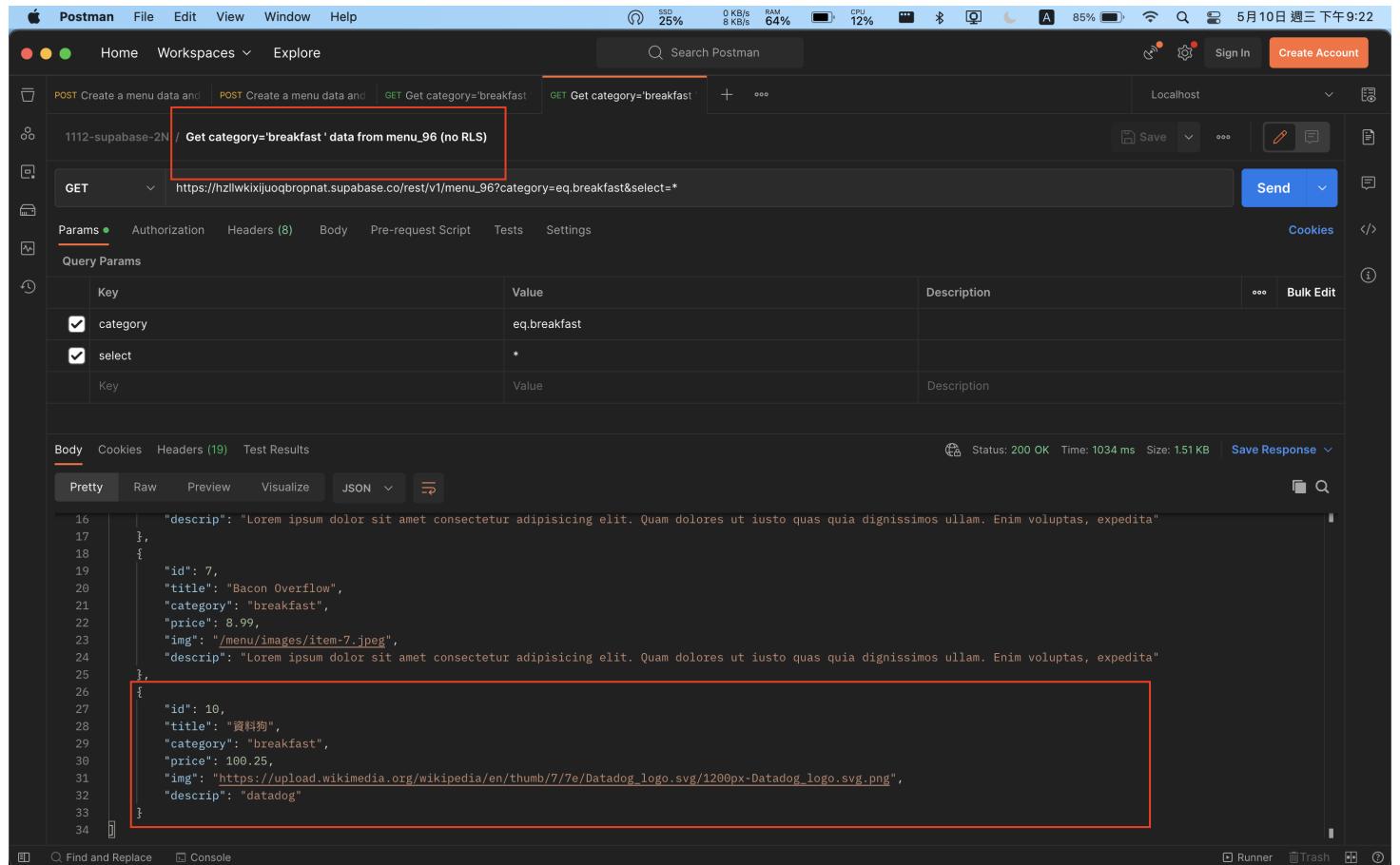
The screenshot shows the Postman application interface. At the top, the title bar includes the Postman logo, File, Edit, View, Window, Help, a search bar labeled 'Search Postman', and system status indicators for SSD (25%), RAM (65%), CPU (10%), battery (93%), and network. The main workspace shows a collection named '1112-supabase-2N / Get category="breakfast" data from menu2_xx (RLS)' with a single GET request. The request URL is https://hzllwixijuoqbropnat.supabase.co/rest/v1/menu2_xx?category=eq.breakfast&select=*&service_role=service_role. The 'Params' tab is selected, showing two parameters: 'category' with value 'eq.breakfast' and 'select' with value '*'. The 'Body' tab displays the JSON response, which includes two items. Item 5 is highlighted with a red box, and item 7 is also highlighted with a red box. The response body is as follows:

```
8     "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam."
9   },
10  {
11    "id": 5,
12    "title": "Egg Attack",
13    "category": "breakfast",
14    "price": 2.99,
15    "img": "/menu/images/item-5.jpeg",
16    "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
17  },
18  [
19    {
20      "id": 7,
21      "title": "Bacon Overflow",
22      "category": "breakfast",
23      "price": 8.99,
24      "img": "/menu/images/item-7.jpeg",
25      "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
26    }
]
```

from menu_xx (no RLS), and from menu2_xx (RLS enabled)

W13-P4: Create a data with category=breakfast, save into menu_xx (no RLS), and from menu2_xx (RLS enabled)

Create a data with category=breakfast, save into menu_xx (RLS not enabled)



The screenshot shows the Postman application interface. A GET request is being made to the URL https://hzlwkixjuoqbropnat.supabase.co/rest/v1/menu_96?category=eq.breakfast&select=*&version=1. The request parameters are set to filter by category='breakfast' and select all fields. The response body is displayed in JSON format, showing two items: one with id 7 and another with id 10. Both items belong to the 'breakfast' category.

Key	Value	Description
category	eq.breakfast	
select	*	

```
16 |     "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
17 | },
18 | {
19 |     "id": 7,
20 |     "title": "Bacon Overflow",
21 |     "category": "breakfast",
22 |     "price": 8.99,
23 |     "img": "/menu/images/item-7.jpeg",
24 |     "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam. Enim voluptas, expedita"
25 | },
26 | {
27 |     "id": 10,
28 |     "title": "資料狗",
29 |     "category": "breakfast",
30 |     "price": 100.25,
31 |     "img": "https://upload.wikimedia.org/wikipedia/en/thumb/7/7e/Datadog_logo.svg/1200px-Datadog_logo.svg.png",
32 |     "descrip": "datadog"
33 | }
```

Postman

File Edit View Window Help

SSD 25% 1 KB/s RAM 65% CPU 9% 85% 5月10日 週三 下午 9:24

Home Workspaces Explore Search Postman Sign In Create Account

POST Create a menu data and GET Get category='breakfast' + ... Localhost

1112-supabase-2N / Create a menu data and save into menu_96 (no RLS)

Save ...

POST https://hzilwkixjuoqbropnat.supabase.co/rest/v1/menu_96

Send

Params Authorization Headers (12) Body Pre-request Script Tests Settings Cookies </> Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 "id": "10",
2 "title": "資料狗",
3 "category": "breakfast",
4 "price": "100.25",
5 "img": "https://upload.wikimedia.org/wikipedia/en/thumb/7/7e/Datadog_logo.svg/1200px-Datadog_logo.svg.png",
6 "descrip": "datadog"
7
```

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize Text

Status: 201 Created Time: 945 ms Size: 474 B Save Response

Find and Replace Console Runner Trash

Create a data with category=breakfast, save into menu2_xx (RLS enabled) by providing service_role token

The screenshot shows the Postman application interface. On the left, the sidebar includes 'Scratch Pad', 'Collections' (with a red box around '1112-supabase-2N'), 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. The main workspace has a 'GET Get category='breakfast'' request highlighted with a red box. The URL is https://hzllwkixijuqbropnat.supabase.co/rest/v1/menu2_96?category=eq.breakfast&select=*. The 'Params' tab shows 'category: eq.breakfast' and 'select: *'. The 'Body' tab displays the JSON response:

```
18 | {
19 |   "id": 7,
20 |   "title": "Bacon Overflow",
21 |   "category": "breakfast",
22 |   "price": 8.99,
23 |   "img": "/menu/images/item-7.jpeg",
24 |   "descrip": "Lorem ipsum dolor sit amet consectetur adipisicing elit. Quam dolores ut iusto quas quia dignissimos ullam.
25 |     Enim voluptas, expedita"
26 | },
27 | {
28 |   "id": 10,
29 |   "title": "資料狗",
30 |   "category": "breakfast",
31 |   "price": 100.25,
32 |   "img": "https://upload.wikimedia.org/wikipedia/en/thumb/7/7e/Datadog_logo.svg/1200px-Datadog_logo.svg.png",
33 |   "descrip": "datadog"
34 | }
```

Postman

File Edit View Window Help

SSD 25% 10 KB/s RAM 65% CPU 9% 85% 5月10日 週三 下午9:25

Home Workspaces Explore

Scratch Pad New Import

POST Create a menu data and ...

1112-supabase-2N / Create a menu data and save into menu_96 (RLS)

Save Localhost

POST https://hzllwkixijuqbropnat.supabase.co/rest/v1/menu2_96

Params Authorization Headers (12) Body Tests Settings Cookies </> Beautify

Body JSON

1 "id": "10",
2 "title": "資料狗",
3 "category": "breakfast",
4 "price": "100.25",
5 "img": "https://upload.wikimedia.org/wikipedia/en/thumb/7/7e/Datadog_logo.svg/1200px-Datadog_logo.svg.png",
6 "descrip": "datadog"
7

Response

Click Send to get a response