DOCUMENTATION NODEJS BACKEND PROGRAM

**Framework EXPRESS**

Node uses the port 8082 to start its process. By default the nodejs route is 8082. That means when you call an endpoint, it basically starts with <https://your_DOMAIN_NAME:8082>.

In order to go and get the endpoint, all created endpoint will be passing through a path name api so that means <https://YOUR_DOMAIN_NAME:8082/api>. This is your main path. Make sure that your frontend developer puts this path. This is the default path.

On this menu contains endpoints for: -

1. Admin users endpoint
2. Customers/merchants endpoint
3. Notification endpoint
4. Transaction endpoint
5. Content endpoint
6. Promo endpoint
7. Banks endpoint
8. Wallet endpoint
9. Settlements endpoint and
10. Merchant link endpoints

Before that there is a thing called an access Permission in the system. Every users have their own limit when it comes to using the api

accessControl(['super\_admin']) // giving permission to internal users

accessPermission(['customer','merchant']) //giving permission to external user

on the above you can see to things stated accessControl function and accessPermission function

Both of this have their own role to play. accessControl belongs or states that it only grants access to the internal users. For example if the permission is granted to the internal users with the role ‘super\_admn’ then you write it like the above code.   
  
For the accessPermission function it belongs only to the external users meaning your customers and merchants. This roles are defined based on the type of users. Basically you have two types of users Customer and Merchant.

p.s Be very careful Nodejs is exteremly case sensitive. When types are selected make sure that they are saved in small letter.

1. **Admin Endpoints**

The admin is a user that control the flows of information with in the system.

Basically there are Four Endpoint that are open endoints

/api/users/signup

/api/users/login

This two endpoints enables admin to signup on the system and also to login

All endpoints except for this two and the other two which are (/api/customers/signup, /api/customers/login) are closed and only accessed by a token generated key.

In each of the endpoint there are many endpoints. Each endpoints handles the basic CRUD operation and also DBRMS

USERS ENDPOINTS lists and what they do?

1. GET METHOD /api/users/active
   1. Shows lists of active users
2. GET METHOD /api/users/
   1. Shows profile of a logged in internal user profile
3. GET METHOD /api/users/logout
   1. Allows user to terminate session
4. GET METHOD /api/users/:id
   1. Shows specific user by id
5. POST METHOD /api/users/login
   1. Allows users to login required fields are
      1. Phone/email as a username
      2. Password
6. POST METHOD /api/users/signup
   1. Allows user to signup on the system
      1. Email
      2. Password
      3. ~~Permissions~~
      4. Role ***[new update]***
7. PUT METHOD /api/:id
   1. Allows user to make an updated to a profile
   2. Filed to be included during updates are: -
      1. Email
      2. Permissions
      3. Roles
      4. Active
8. DELETE METHOD /api/:id
   1. Only super admin can delete user
9. **Customer’s Endpoints**
10. GET METHOD /api/customes
    1. Showing all list of customers and merchants information
11. GET METHOD /api/customers/profiles
    * 1. ALLOWING LOGGED IN USER TO SEE HIS/HER OWN PROFILE
12. GET METHOD /api/customers/transaction/:id
    * 1. Getting lists of transaction made by a specific user
13. POST METHOD /api/customers/search
    * 1. Enabling admins to search for customers or merchants by their email or mobile
14. POST METHOD /api/customers/document/update
    * 1. Enabling customers or merchants to update their own document which is found on the database table kyc
15. PUT METHOD /api/customers/update\_profile
    * 1. Enables external users to update thir profile page
16. PUT METHOD /api/customers/update\_password
    * 1. Enables external users to updae their own password
17. PUT METHOD /api/customers/:id
    * 1. Enables the admin to update the profiles of the external users
18. DELETE METHOD /api/customers/:id
    * 1. Enables the external user which is either the control of admin or super admin to remove role
19. **Content Endpoints**
    1. GET METHOD /api/contents/
       1. Enables admin to see lists of contents
    2. GET METHOD /api/contents/:id
       1. Enables admin to see specific content by using the id which is sn
    3. POST METHOD /api/contents
       1. Enables admin to add content. Lists of paramer to app are
          * 1. Data\_key
            2. Data
    4. PUT METHOD /api/contents/:id
       1. Enables admin to update a specific contents
    5. DELETE METHOD /api/contents/:id
       1. Enables the admin to delete a content
20. **Notifications Endpoints**
    1. GET METHOD api/notification
       1. Enables to see all lists of notification
    2. GET METHOD api/notification/lists
       1. Enables cutomers to see their own notification
    3. GET METHOD api/notification/:id
       1. Enables to see a specific notification by the admin
    4. POST METHOD api/notification
       1. Enables to create a notification
       2. Parameter to pass in order to create is
          1. User, msg, redirect and extra
    5. PUT METHOD api/notification/:id
       1. Update the notification by calling specific id
    6. DELETE METHOD api/notification/:id
       1. REMOVES NOTIFICATION
21. **Promotion ENDPOINTS**
    1. GET METHOD api/promo => view all promo
    2. GET METHOD api/promo/:id => view specific promo
    3. POST METHOD api/promo => add promo
       1. Parameter to pass is
          1. Image, redirect, active, created\_at
       2. Also upload the file for the image
    4. PUT METHOD api/promo/:id = > allows to edit the tests which is
       1. Redirect, active
    5. PUT METHOD api/promo/update\_image/:id
       1. Enables the user to update the profile photo of the promo incase a mistake occurs
       2. Parameter to pass
          1. Image
    6. DELETE METHOD api/promo/:id
       1. Removes the promo
22. **Banks ENDPOINTS**
    1. GET METHOD /api/banks/ => showing lists of banks
    2. GET METHOD /api/banks/:id => specific banks
    3. GET METHOD /api/banks/customers => external users enables to see their bank account
    4. POST METHOD /api/banks/
       1. Enables users to add bank account profiles
       2. External users will create this bank account
       3. Parameter to pass are
          1. User\_id, bank\_name, account\_number, iban, bic, is\_default
    5. PUT METHOD /api/banks/:id
       1. Enables admin to update the bank profile of a users
       2. Parameter to pass is
          1. bank\_name, account\_number, iban, bic, is\_default
    6. DELETE METHOD /api/banks/:id => remove the bank account by the admin
23. **WALLETS ENDPOINTS**
    1. GET METHOD /api/wallets => show all lists of wallets
    2. GET METHOD /api/wallets/:id => show specific wallets
    3. POST METHOD /api/wallets =>
       1. Customers allowed to add wallet of their own
          1. Parameter to pass are: - Amount and type
    4. PUT METHOD /api/wallets/:id => update specific wallets
    5. DELETE METHOD /api/wallets/:id => removes wallets
24. **SETTLEMENTS ENDPOINTS**
    1. GET METHOD /api/settlement => show lists of settlement
    2. GET METHOD /api/settlement/:id => show specific settlement
    3. POST METHOD /api/settlement
       1. Enables customers to add settlements
       2. Parameters to pass are
          1. amount, account\_id, and reasons
    4. PUT METHOD /api/settlement/:id => update
    5. DELETE METHOD /api/settlement/:id => remove
25. **Merchant Link Endpoints**
    1. GET METHOD api/merchant\_links = > show all
    2. GET METHOD api/merchant\_links/:id => specific id view
    3. GET METHOD api/merchant\_links/showLists => enables only the merchant to see lists of data
    4. POST METHOD api/merchant\_links
       1. Only customer will create a merchant link
       2. Parameters to pass is id, merchant, amount, reason
    5. PUT METHOD api/merchant\_links/:id => update specific
    6. DELETE METHOD api/merchant\_links/:id => remove specific
26. **Transaction ENDPOINTs**

This part of the endpoint is a bit complicated.

* 1. GET METHOD /api/transaction/customers =
     1. Enables the customers to see lists of transaction made
  2. GET METHOD /api/transaction
     1. Enables the admin to see all transaction
  3. GET METHOD /api/transaction/:id
     1. Enables to select specific transaction
  4. POST METHOD /api/transaction/
     1. This part of the documentation is a bit complicated because transaction of money is involved
     2. Only external users get to access this endpoints because they are the one that does the transaction inorder for their wallet to be updated
     3. Parameter to pass are
        1. Amount
        2. Type
        3. Extra\_info
        4. Trans\_type

The type that the user selects depends

Type 1: adding wallet to own wallet

Type 2: money transfer

Type 3: money receive to merchant

In type 3 is selected must also add a parameter recieverID which is merchant id

P.S => Incase something went wrong in the transaction history table then every other transaction will stop and restore to its previous.

P.s=> you can make an update on the source code as to who get to access a certain endpoints by going to your routes folder.

**Updates**

Internal Users account has been updated: -

1. Permissions has been removed
2. Role has substituted permission

On the role side

* You are only allowed to enter one string not multiple string
* Allowed strings are
  + super\_admin
  + content
  + banner
  + customer
  + staff

Please give a single name for those with multiple access.   
For example for those with the roles of banner, content and staff => give a new name

Logout has been updated. Now internal and external users can logout successfully.

The token expires every 30 minutes and automatically destroys the authenticated token.

P.s you can modify the routes folder of each files to define or identify who gets to access.