Paas project

Produced by G.O.4 Team

Purpose and summary

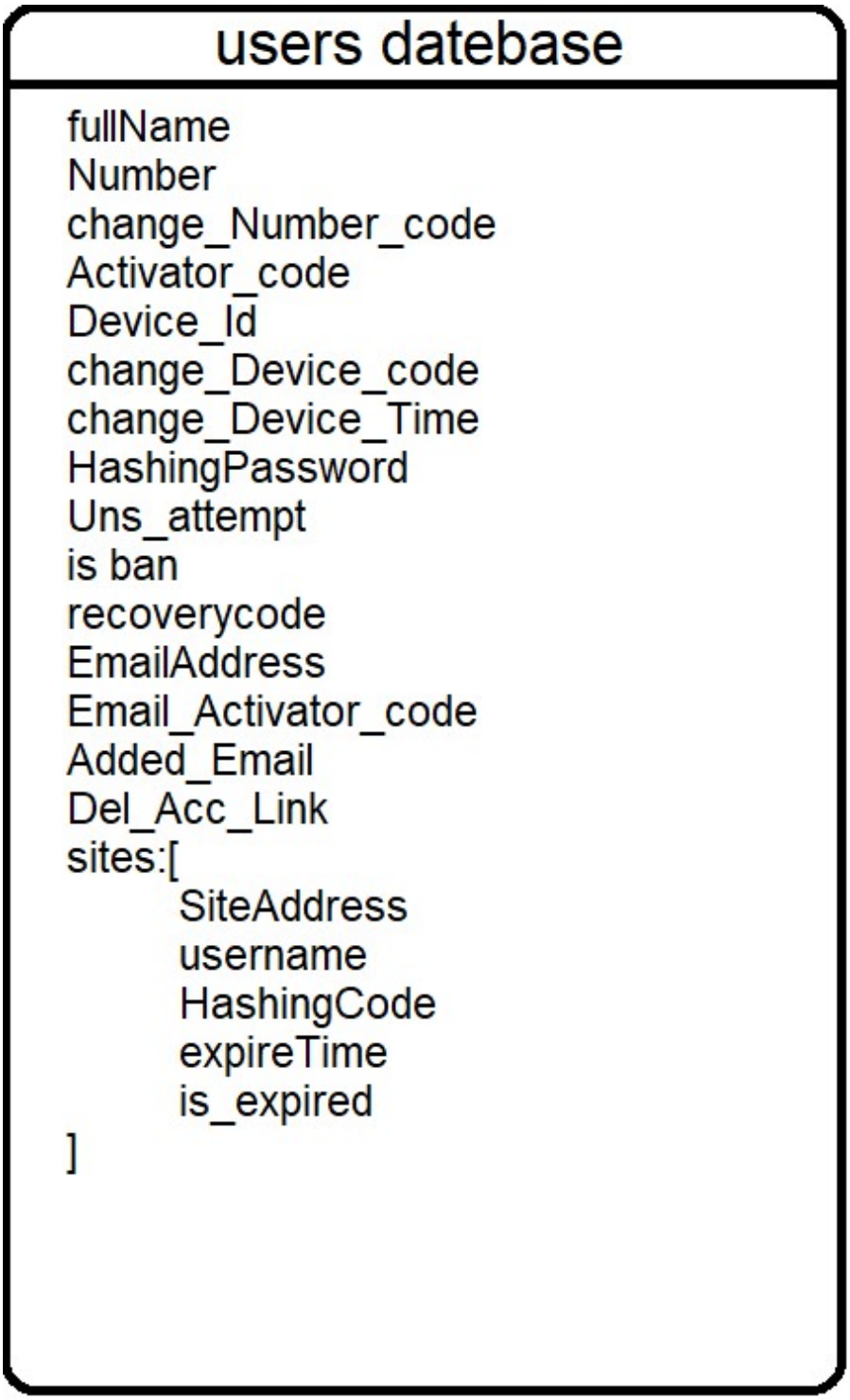
The purpose of this program is to use the second password to increase the security of loggining the sites.

In this way, after logging in, you log in to your paas account in the application and receive the second password from the box of the site, and enter the second password in the password request section.

All passwords are stored hashed on the site so that passwords will not be accessible if anonymous people access them.

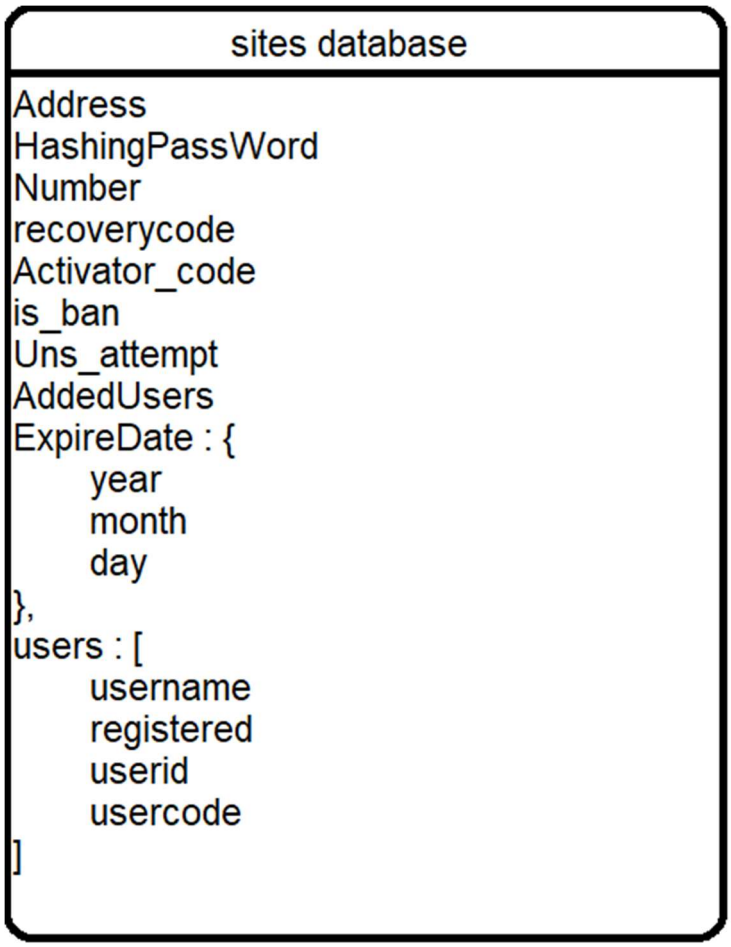
DateBase

User Datebase



- fullName: user FullName
- Number: user phone Number
- Change_Number_code: a code that use for changing phone Number
- Activator_code: a code that use for Activation Account after registration or ban
- Device_Id: each Account need to be Active on special Device and cant and it is Device Id of that Device
- change_Device_code: a code that use for changing Device of Account
- change_Device_Time: each change_Device_code is active for 10 minute
- HashingPassword: The password is stored as a hash
- Uns_attempt: each unsuccesfull try to login, increases this variable by one unit
- is_ban: true is ban and false is not ban
- recoverycode: the code that use for recovery Account for Activation
- EmailAddress: what I say ? 😐
- Email_Activator_code: for Activation of Email use this code
- Added_Email: true if Email is Active and false if Email is not Active
- Del_Acc_Link: it's a code that use for Delete Account without login and just with Email
- SiteAddress: what I say too? 😐
- username: a username that use user at that site
- HashingCode: a code that use for login (main code)
- expireTime: each code has 1 minute time to use
- is_expired: is code is expire or not?

Sites Database



Fields without user databases fields:

- Address: site Address
- AddedUsers: each site can Add previous users one time
- year & month & day: expire Account time
- username: username of user at this site
- registered: true is This site Account Added to user Account
- userId: phone Number of user
- usercode: the code that use for Adding site to user Account

How to work

Signup and Activation Account

1-enter Full Name and Phone Number and password

2-click on link that sent to phone Number

Add site to user Account

1-signup to a site

2-get entirely code from that site (with QR code or manual)

3-enter code, username for that site and site Address (if enter with QR code, all of them is automatic)

Get and confirm code

1-Login to a site

2-Login to paas Application and get code from that site box

3-Write code in the password request section and send it

Delete Account with Email

1-enter Email

2-confirm it from clicking on the link that sent to Email

3-enter Email Address in the Email section on the site

4-click on link that sent to Email

Change Device

1-login to your account

2-click on ”change device”

3-scan QR code from new Device

Change Number

1-login to your Account

2-click on “change Number”

3-input new Number

4-confirm Number with clicking on the link that sent to new Number

User Signup

Method: post

Address: localhost:4000/signup

Input fields:

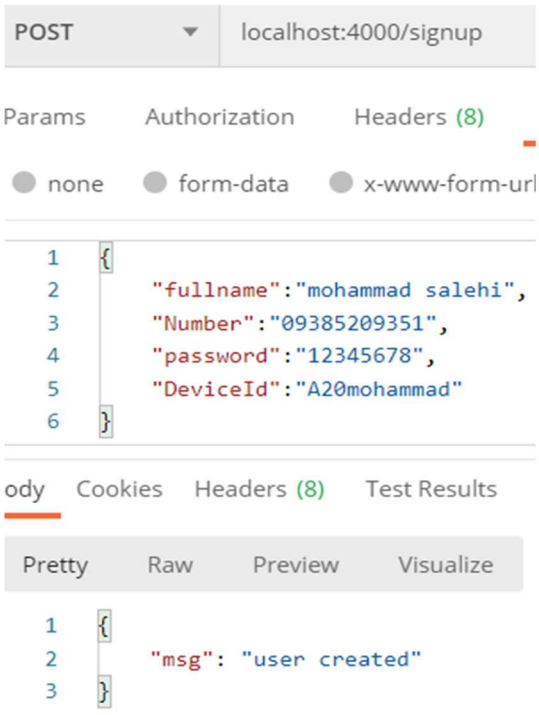
- fullname
- Number
- DeviceId
- Password

Rules:

- The DevideId is not received from the user and is read automatically from device

Responses:

- res.status(403).json({msg:'Unsuccessful'}) : Unsuccessful 😐
- res.status(403).json({msg:'this Number is already exist'}) : the Number is repetitive and is already exist
- res.status(403).json({msg:'Wrong Phone Number format'}) : the phone Number format is wrong(first Number must be 0 and its length must be 11 Numbers)
- res.status(403).json({ msg:'enter true fullname'}) : the name most not be empty
- res.status(403).json({msg:'Enter longer password'}) : password must be 8 characters or more
- res.status(200).json({msg:'user created'}) : successful



User login

Method: post

Address: localhost:4000/login

Input fields:

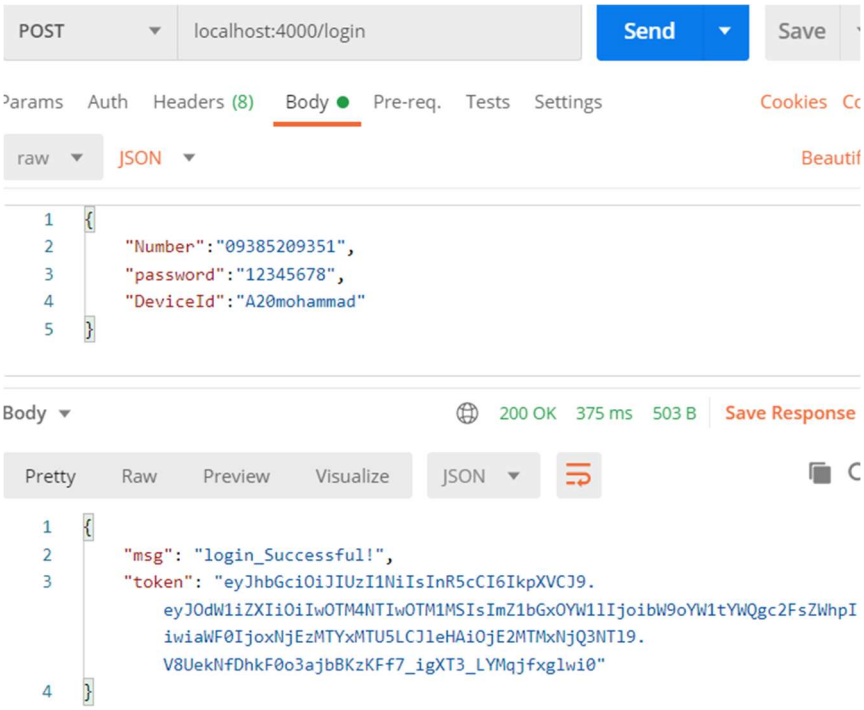
- Number
- DeviceId
- Password

Rules:

- The DevideId is not received from the user and is read automatically from the device And if it is different from the previous one, The server does not give tokens.

Responses:

- res.status(403).json({msg:'Wrong_Device_ID'}) : DeviceId is different from previous DevideId.
- res.status(401).json({msg:'Auth Failed!'}) : something is wrong.
- res.status(403).json({msg:'Unsuccessful'}) : Unsuccessful
- res.status(403).json({msg:'ban'}) : Account is ban.
- res.status(200).json({
  msg:'login_Successful!',
  token:token
  }) : login is Successful and Tokens given in the Tokens field.



Active User Account

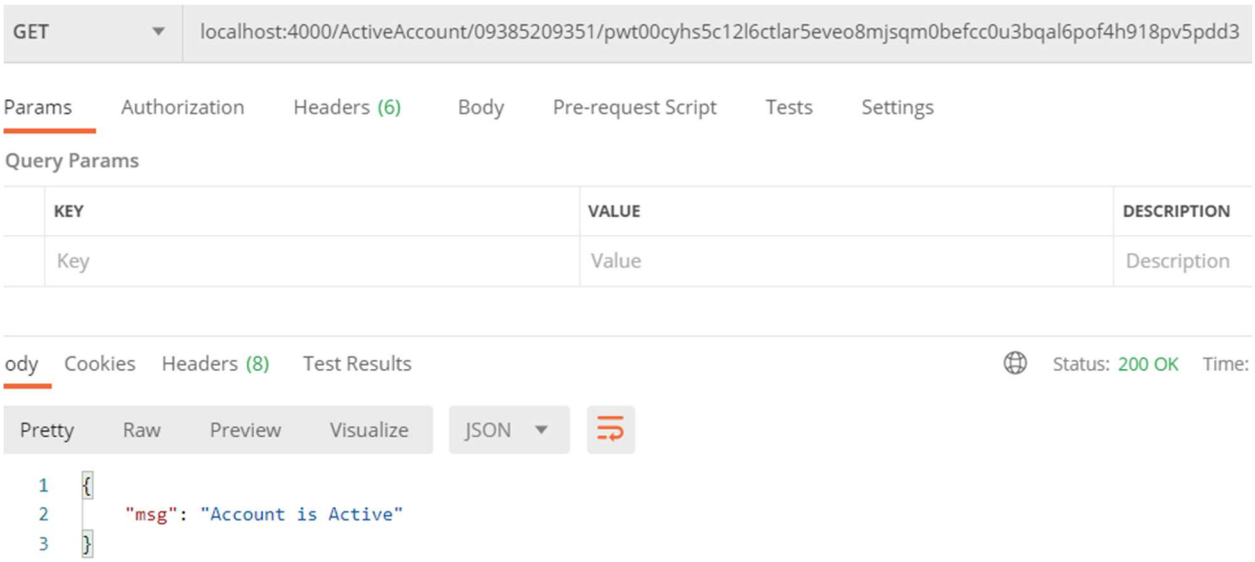
Account need to be actived After signing up and become ban.

Method: get

Address: localhost:4000/Number(user Number)/code(sent to phone Number)

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'Account is Active'})



Add Site to User Account

The user must first be added by the same site.

Method: post

Need the Token

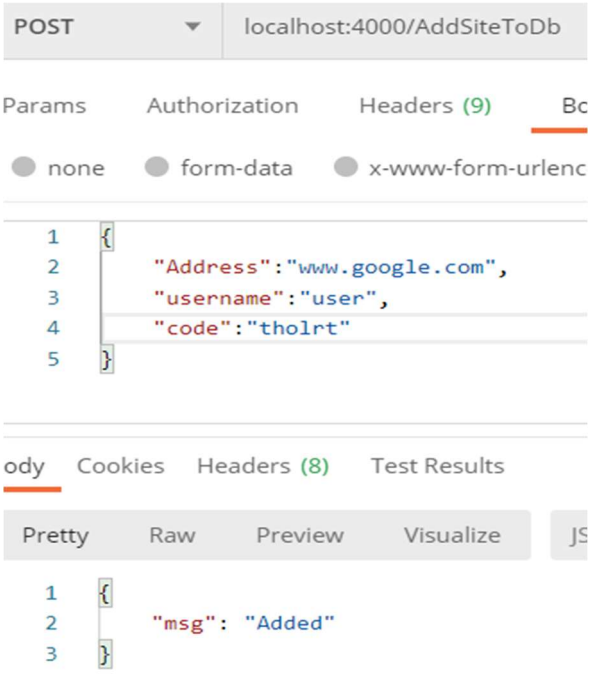
Address: localhost:4000/ AddSiteToDb

Input fields:

- Address
- username (User username on the same site)
- code (give from site and sent by QRcode or manual input)

responses:

- res.status(404).json({msg:'need_add_to_db_from_site'}) : The site has not Added that user to DataBase
- res.status(403).json({msg:'you are already added this site'}) : the site is already added
- res.status(403).json({msg:'wrong code'}) : the code is wrong
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'Added'})



Get the password

Method: get

Need the Token

Address: localhost:4000/getcode

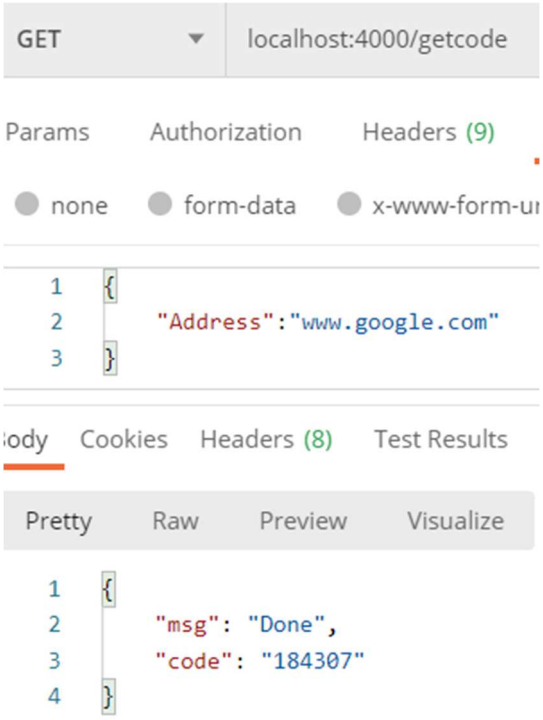
Input fields:

- Address (The site that the user wants to login)

Responses:

- res.status(404).json({msg:'Wrong_Address'}) : site Address is wrong
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({
  msg:'Done',
  code:code

  })



Send recovery link

When Account is banned need to use it

Method: post

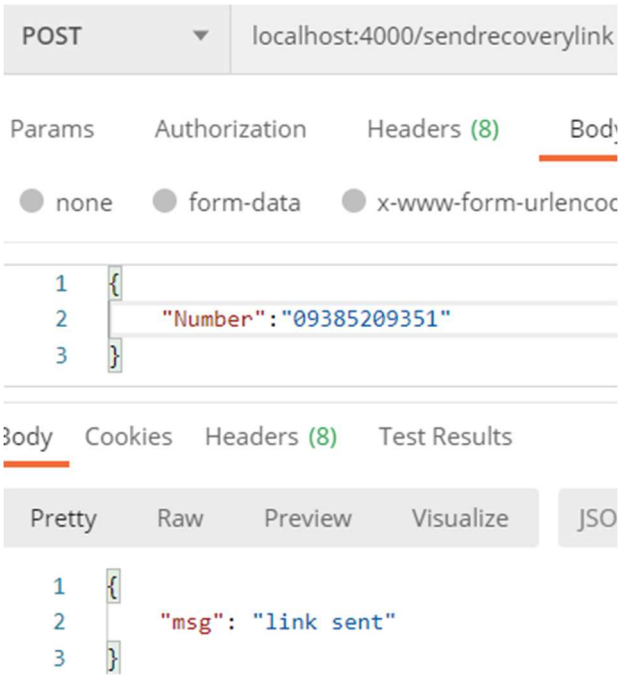
Address: localhost:4000/sendrecoverylink

Input fields:

- Number (user Number)

Responses:

- res.status(403).json({msg:'wrong Number'})
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'link sent'}) : link sent to phone Number



Get user Account Information

Method: get

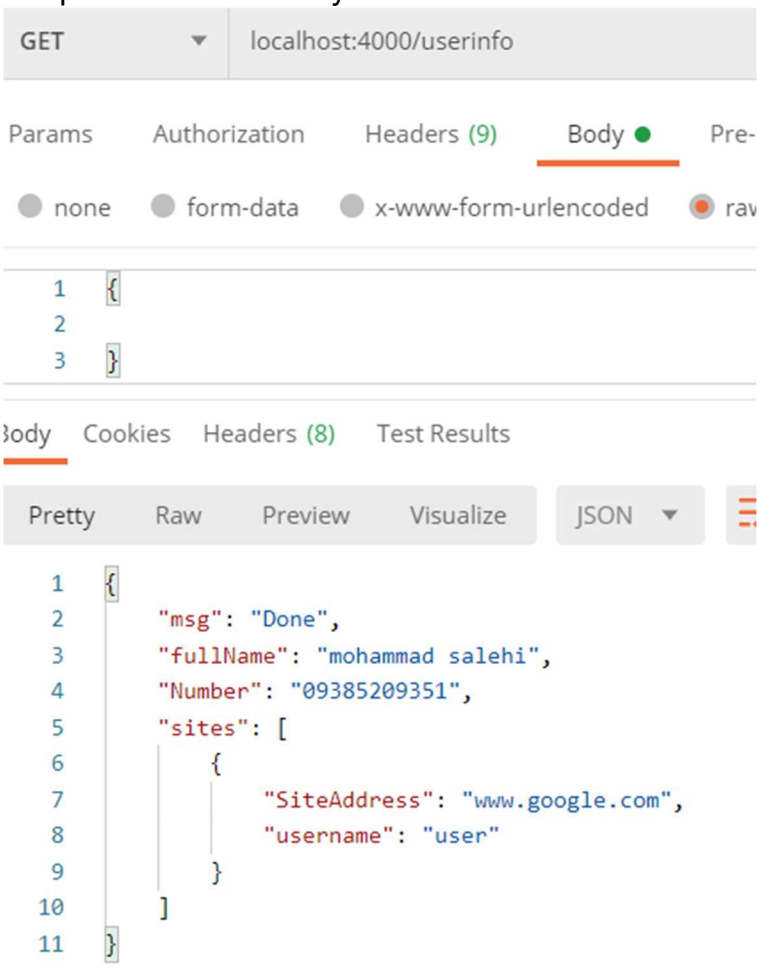
Need the Token

Address: localhost:4000/getcode

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({
  msg:'Done',
  fullName:fullName,
  Number:Number,
  sites:postsites
  })

*** postsites is an Array that take a list of sites that user Added to his Account



Delete Account

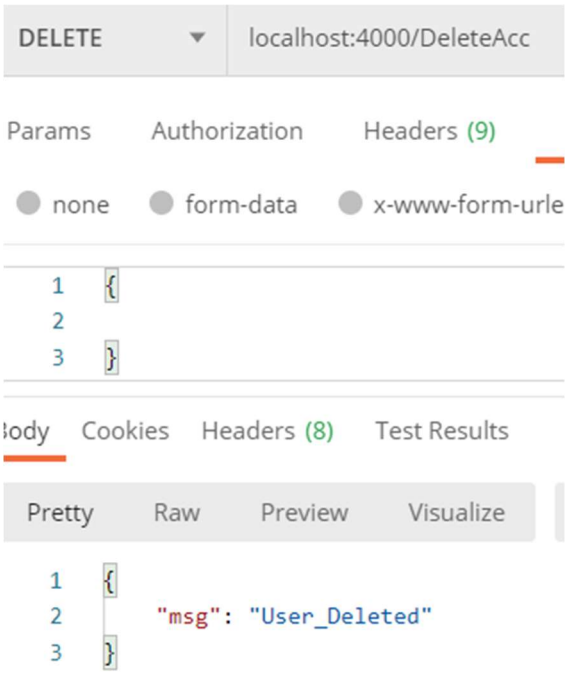
Method: delete

Address: localhost:4000/DeleteAcc

need Token

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'User_Deleted'})



Add Email to Account (send link)

Method: post

Need the Token

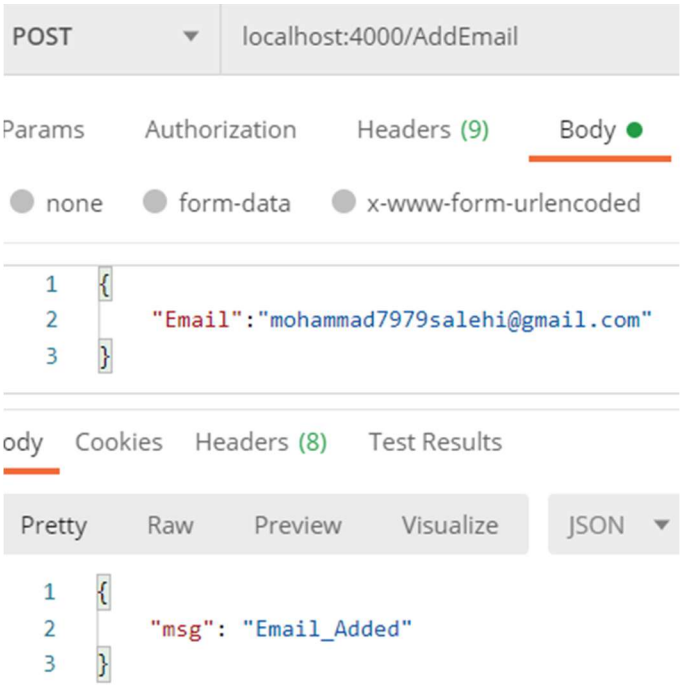
Address: localhost:4000/AddEmail

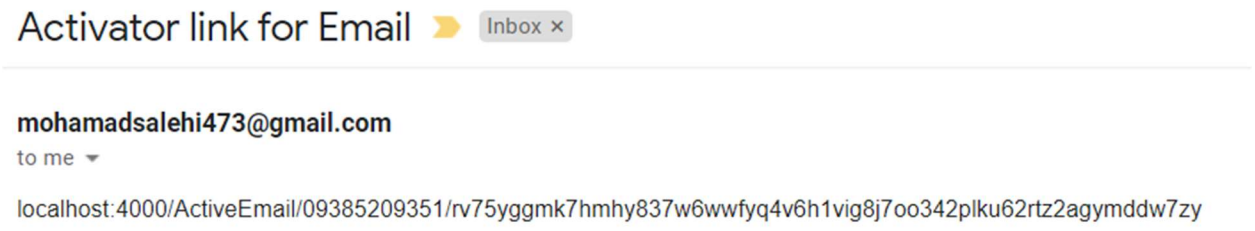
Input fields:

- Email

Responses:

- res.status(403).json({msg:'Wrong_Email'}) : Email format is wrong
- res.status(200).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'Email_Added'})





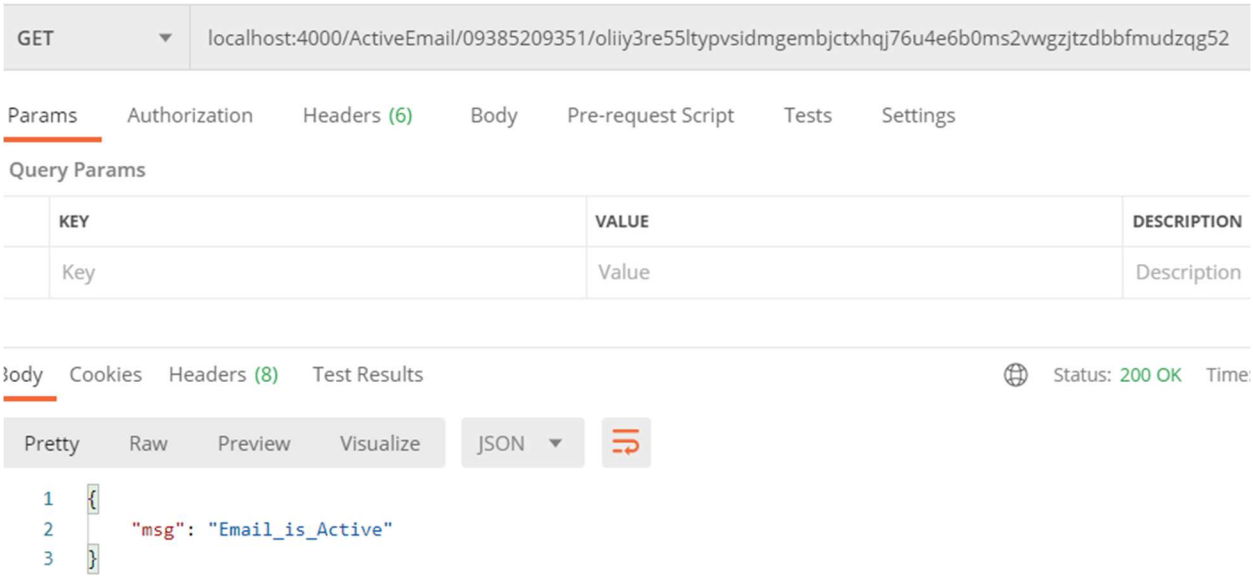
Active Email

Method: get

Address: localhost:4000/ActiveEmail/(user Number)/(user code)

Responses:

- res.status(403).json({msg:'Wrong_link'}) : wrong code
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'Email_is_Active'})



Show QR Code (for change Device Id)

Method: get

Need the Token

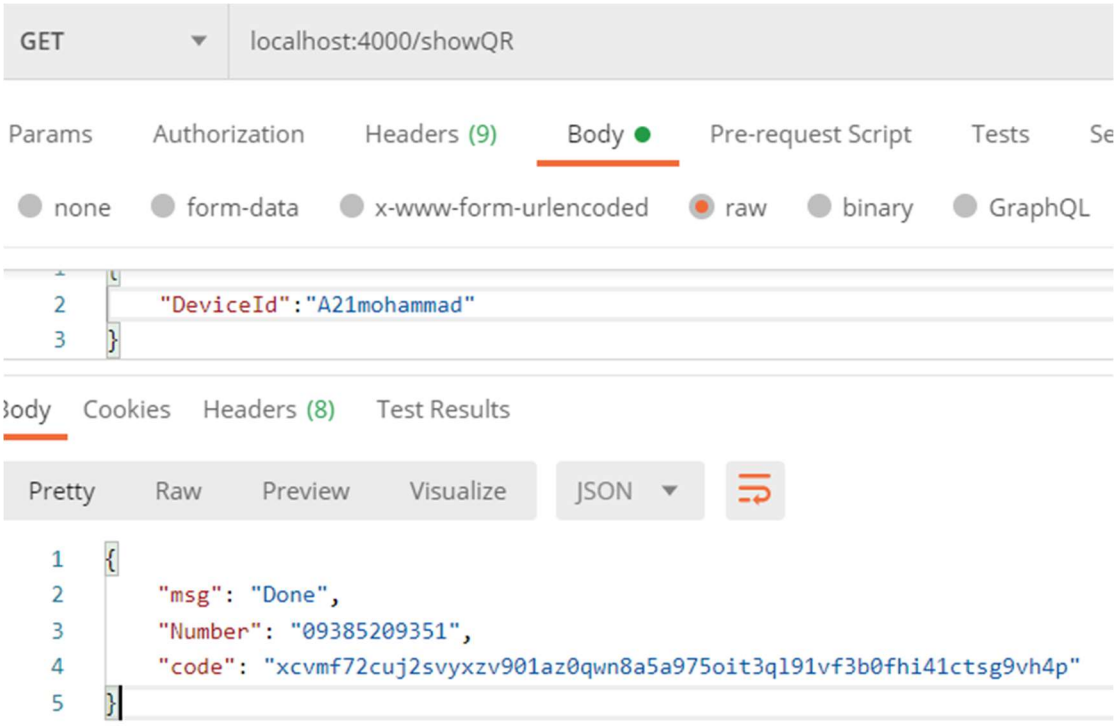
Address: localhost:4000/showQR

Input fields:

- DeviceId

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({
  msg:'Done',
  Number:Number,
  code:code
  }) (show this info as QRCode)



Get QRCode for change DeviceID

Method: post

Need the Token

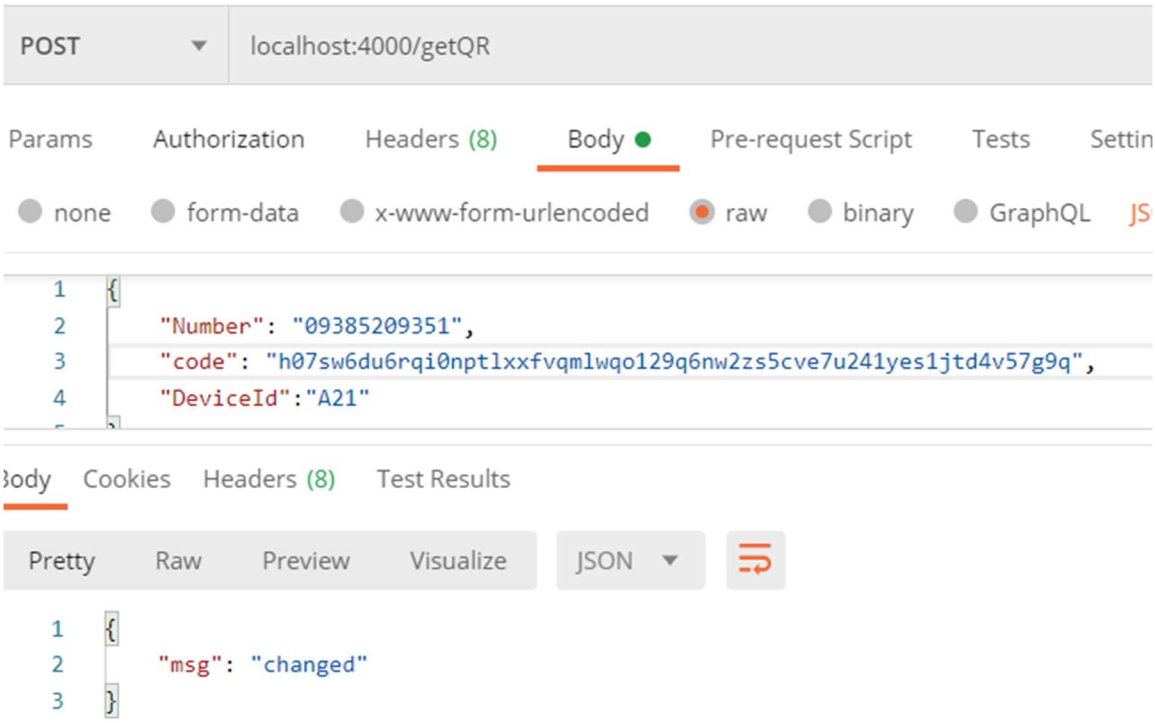
Address: localhost:4000/getQR

Input fields: (automatically)

- Number (Account phone Number)
- code ()
- DevideId

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'changed'})



Get link for change Number

Method: post

Need the Token

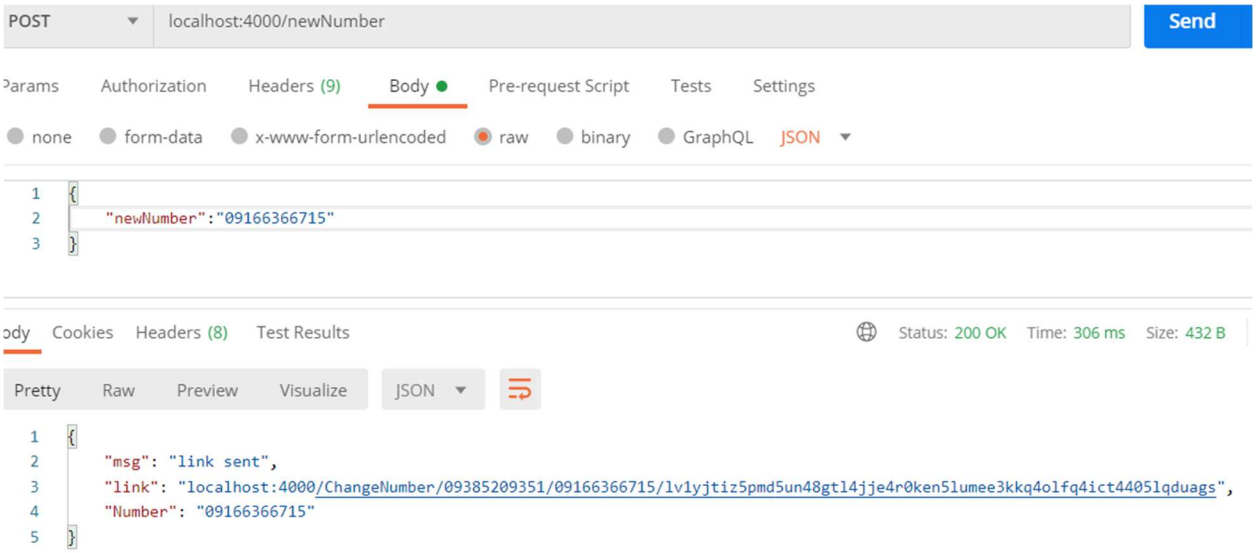
Address: localhost:4000/newNumber

Input fields:

- newNumber

responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'link sent'})



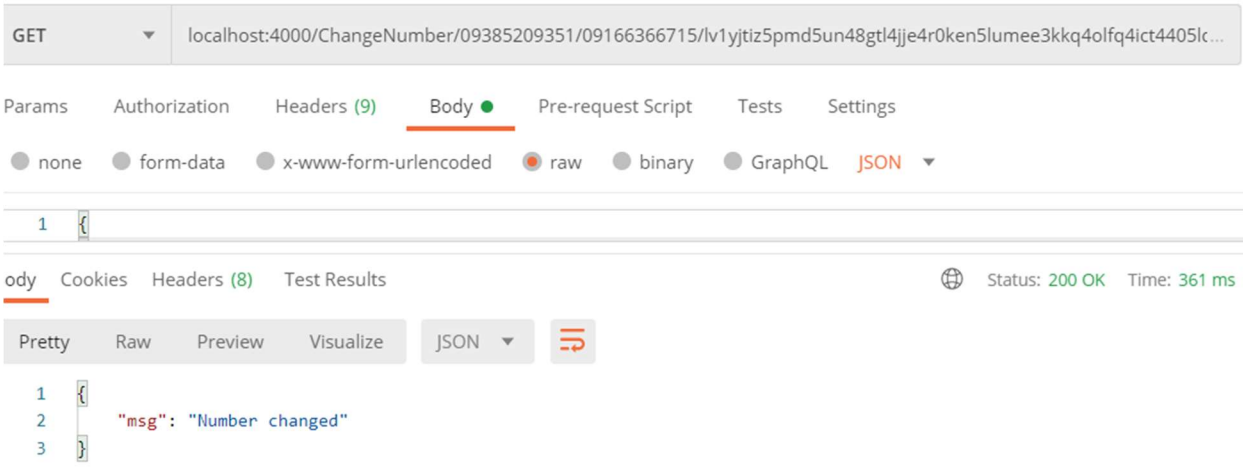
Change Number with link

Method: get

Address: localhost:4000/(Account Number)/(new Number)/(code)

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'Number changed'})



Recovery password

Method: get

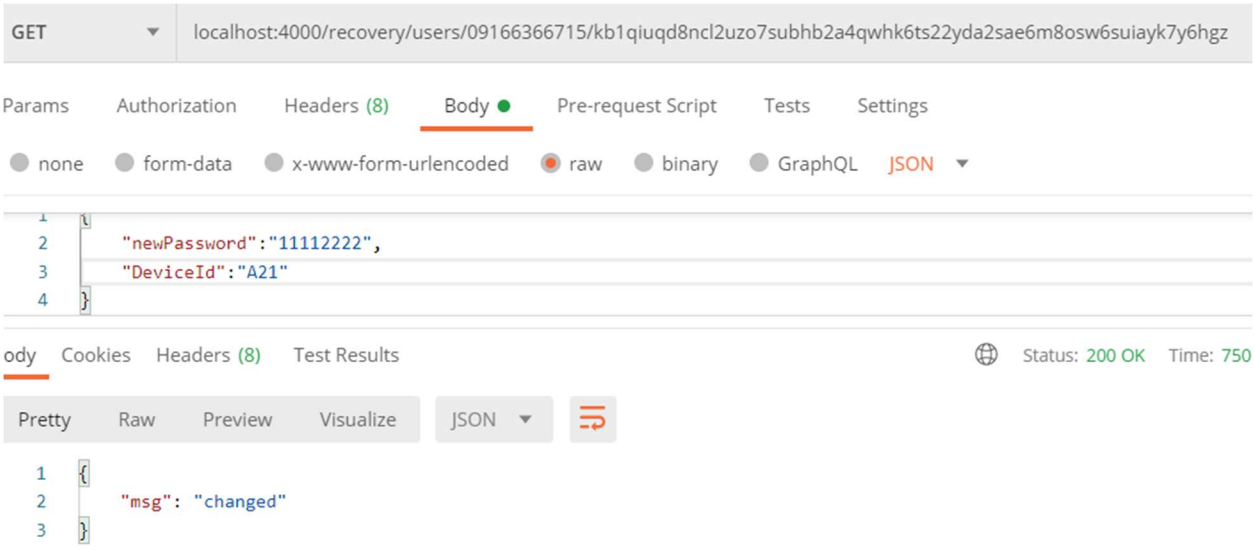
Address:localhost:4000/(mode (user or site))/(ID (user number or site Address))/(code)

Input fields:

- newPassword
- DeviceId

Responses:

- res.status(200).json({msg:'please choose longer password'})
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'changed'})



Site Registration

Method: post

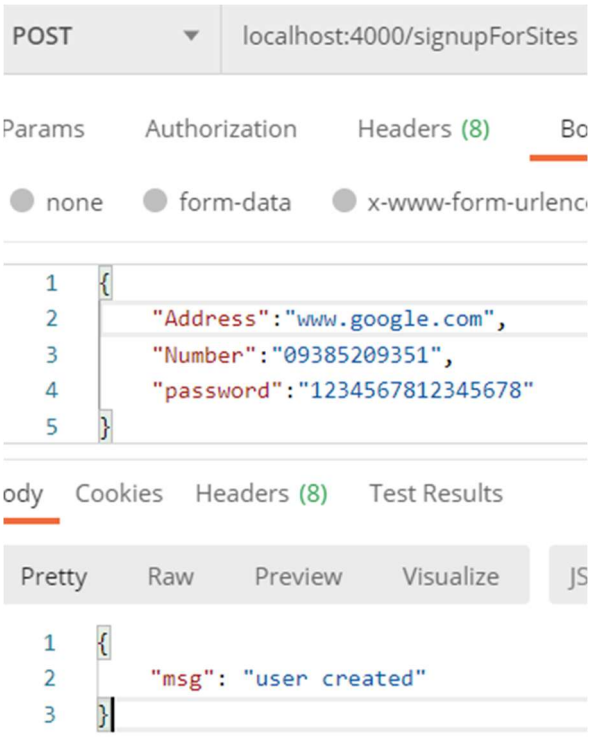
Address: localhost:4000/siteregistration

Input fields:

- Address
- Number
- Password

Responses:

- res.status(403).json({msg:'Enter longer password'})
- res.status(403).json({msg:'Wrong Address'})
- res.status(403).json({msg:'Wrong Phone Number format'})
- res.status(403).json({msg:'this Number is already exist'})
- res.status(403).json({msg:'this Address is already exist'})
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'user created'}) (need to active account with link)



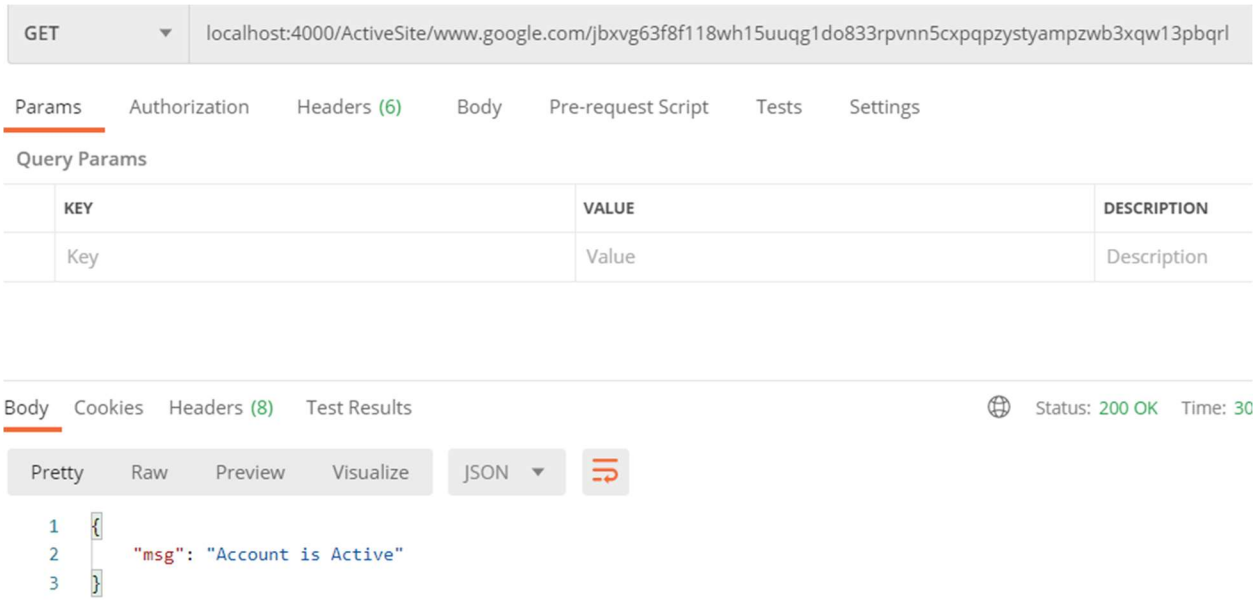
Active site Account

Method: get

Address: localhost:4000/ActiveSite/(site Address)/(code)

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'Account is Active'})



Site login

Method: post

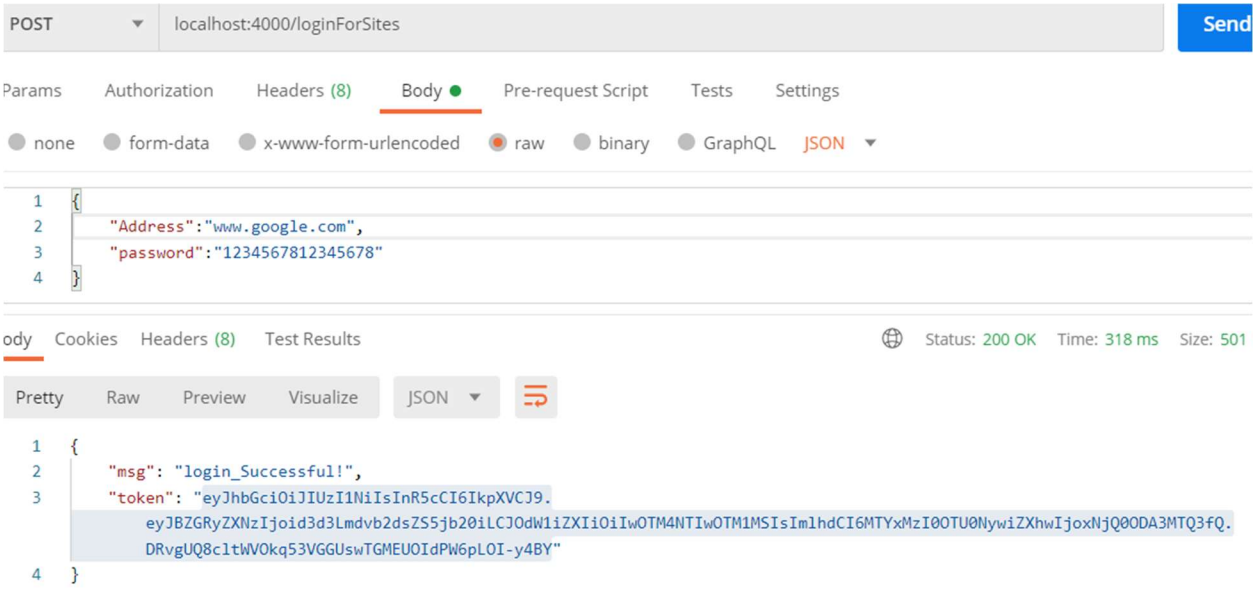
Address: localhost:4000/loginForSites

Input fields:

- Address
- password

Responses:

- res.status(401).json({msg:'Auth Failed!'})
- res.status(403).json({msg:'ban'})
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({
  msg:'login_Successful!',
  token:token
  })



Add users to site database

Method: post

Need the Token

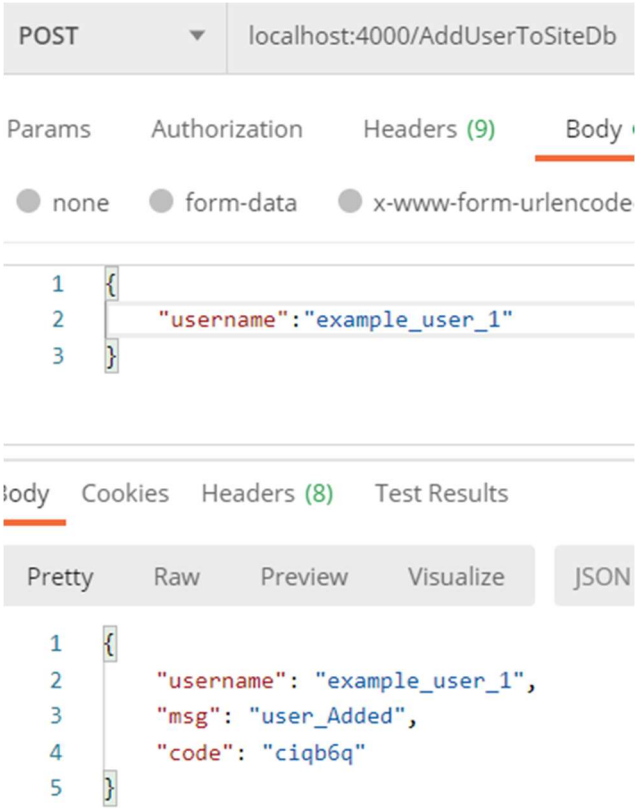
Address: localhost:4000/AddUserToSiteDb

Input fields:

- username (user username for that site)

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(403).json({
  username:username,
  msg:'already_Added_from_before'
  })
- res.status(200).json({
  username:username,
  msg:'user_Added',
  code:code
  })



Confirm code

Method: get

Need the Token

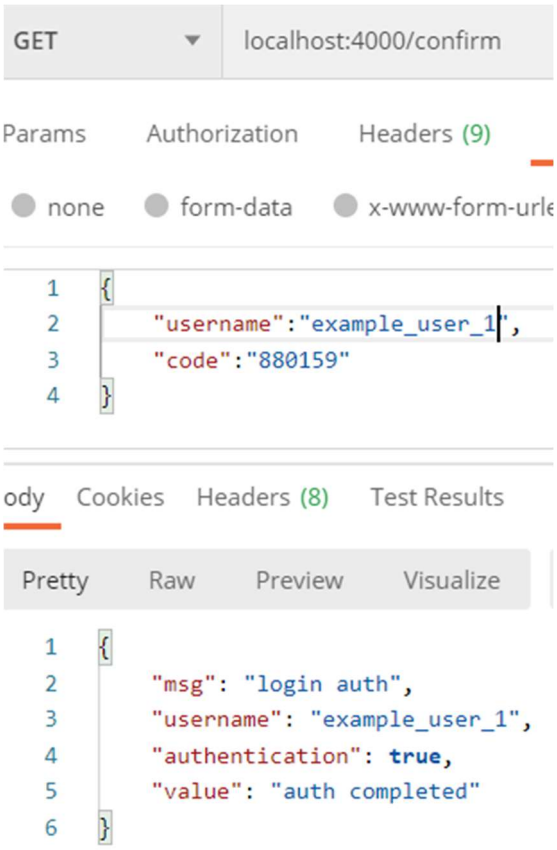
Address: localhost:4000/confirm

Input fields:

- username
- code

Responses:

- res.status(404).json({msg:'wrong username'})
- res.status(404).json({msg:'wrong Address'})
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({
  msg:'login auth',
  username:username,
  authentication:false,
  value:'wrong code'
  })
- res.status(200).json({
  msg:'login auth',
  username:username,
  authentication:false,
  value:'expire time error'
  })
- res.status(200).json({
  msg:'login auth',
  username:username,
  authentication:false,
  value:'has been used before'
  })
- res.status(200).json({
  msg:'login auth',
  username:username,
  authentication:true,
  value:'auth completed'
  })



Send recovery link for sites

Method: post

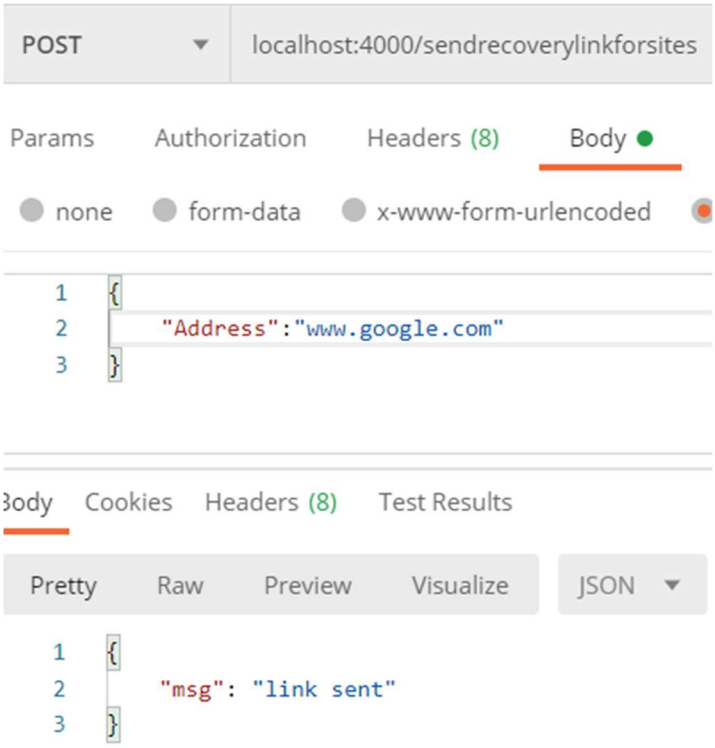
Address: localhost:4000/sendrecoverylinkforsites

Input fields:

- Address

Responses:

- res.status(403).json({msg:'wrong Address'})
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'link sent' })



Sites Account renewal

Address: localhost:4000/renewal

Method: post

Need the Token

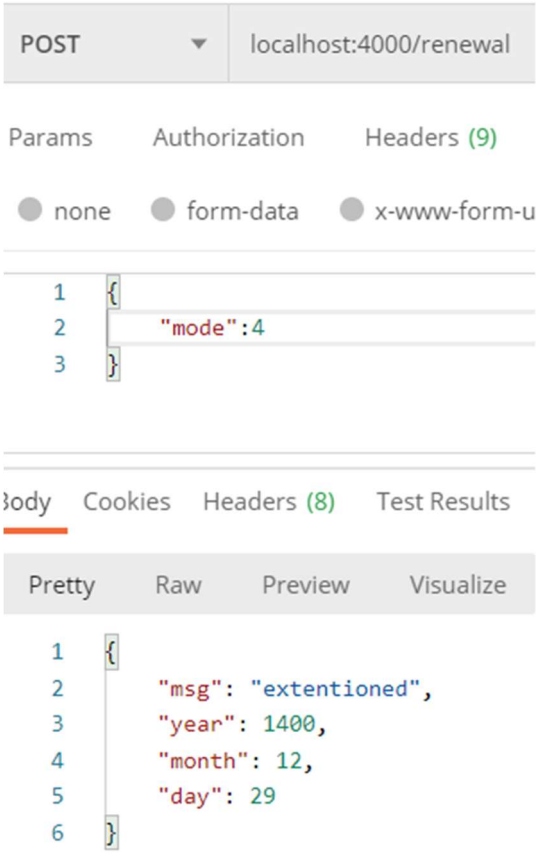
Input methods:

- mode (1= 1 month, 2= 3 month, 3= 6 month, 4= 1 year)

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({
  msg:'extentioned',
  year,
  month,
  day

  })



Add previous users

Method: post

Need the Token

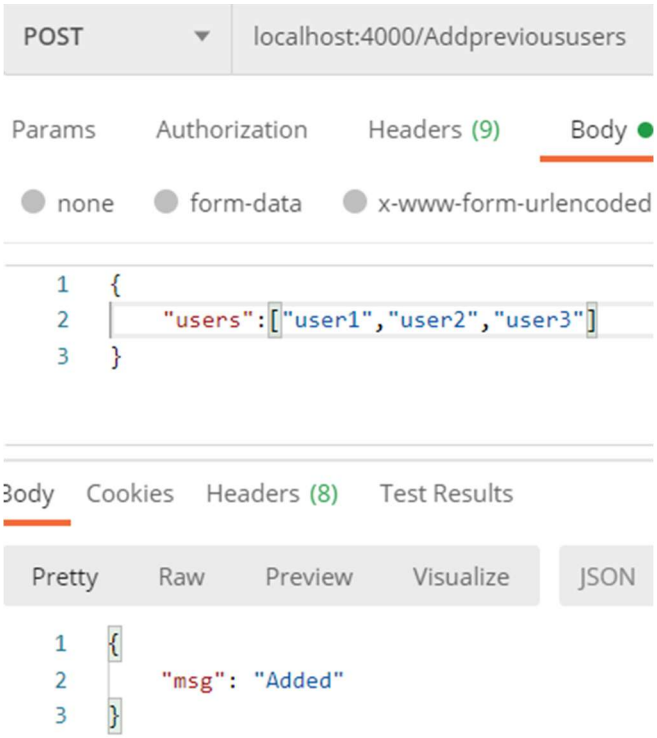
Address: localhost:4000/Addprevioususers

Input fields:

- users(Array of usernames)

Responses:

- res.status(403).json({msg:'already Added'}) : that site already added users
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'Added'})



Dashboard informations

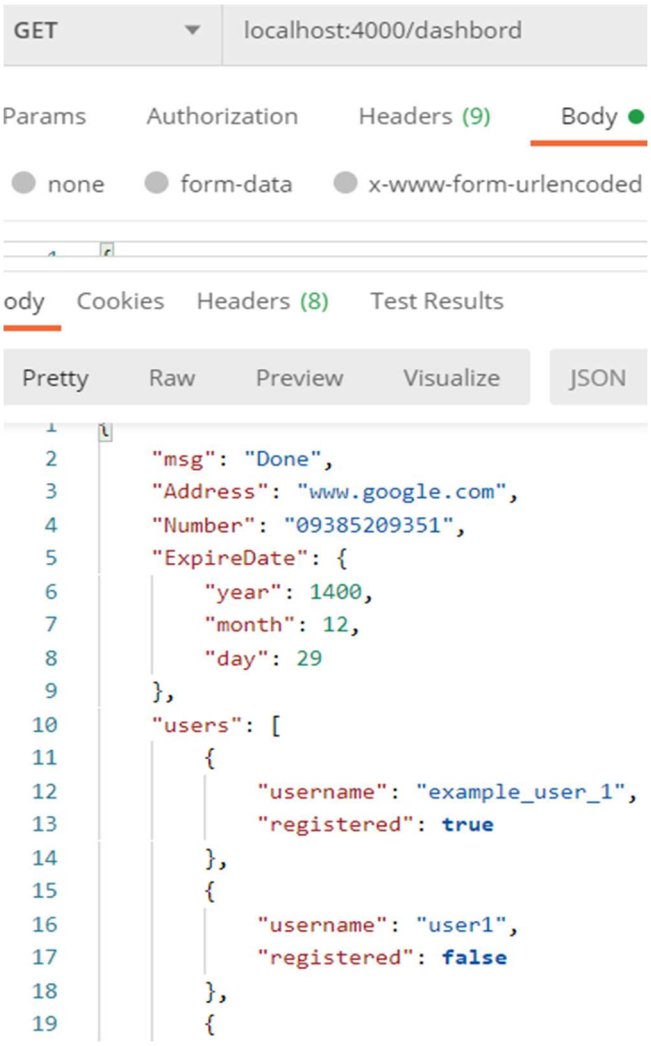
Address: localhost:4000/dashboard

Method: get

Need the Token

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({
  msg:'Done',
  Address,
  Number,
  ExpireDate,
  users (Array)
  })



Get user code

Method: get

Need the Token

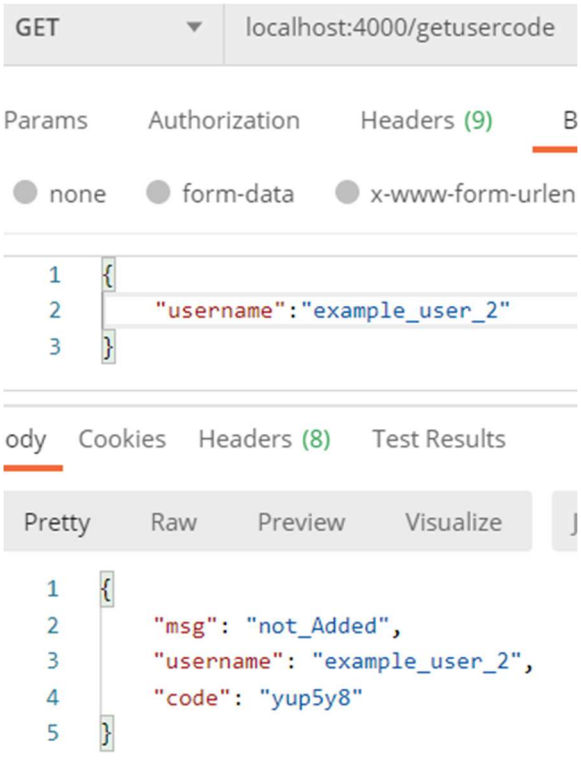
Address: localhost:4000/getusercode

Input fields:

- username

Responses:

- res.status(404).json({msg:'user_not_founded'})
- res.status(403).json({msg:'Unsuccessful'})
- res.status(400).json({msg:'already_added'})
- res.status(200).json({
  msg:'not_Added',
  username,
  code
  })



Send delete Account link to Email

Method: post

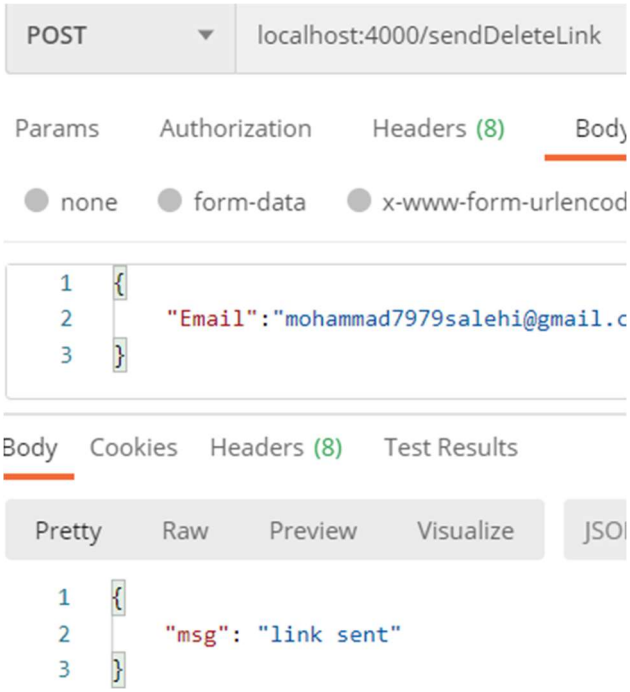
Address: localhost:4000/sendDeleteLink

Input fields:

- Email

Responses:

- res.status(403).json({msg:'Email is not Active'})
- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'link sent'})



Delete Account by link

Method: get

Address: localhost:4000/DelByLink/(Email)/(code)

Responses:

- res.status(403).json({msg:'Unsuccessful'})
- res.status(200).json({msg:'User_Deleted'})