\*\*\*Chú ý quan trọng: vì database server đặt trên cloud, tôi dùng knex để truy cập nên dôi khi nó sẽ lag (có thể do cả mạng nữa), nếu báo lỗi server thì có thể thử request lại lần nữa (do knex lỗi nên nó bị thế, sau này triển khai backend lên cloud là hết)

1. Login

- Mục đích: đăng nhập vào hệ thống với tài khoản của người dùng

- Phương thức: POST

- URL: localhost:3001/auth/login

- Tham số: gửi trong body

//có thể gửi username hoặc email là tuỳ, nhưng key phải là "username"

+ Ví dụ đăng nhập bằng tên đăng nhập

{

    "username": "admin01",

    "password":"huython"

}

+ Ví dụ đăng nhập bằng email

{

    "username": "vuquanghuy21081999@gmail.com",

    "password":"huython"

}

- Dữ liệu trả về:

{

    "success": **true**,

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDQwMTkwNzA4MTIwNjEwMDA0NTkiLCJpYXQiOjE2Mzc4NDUyNDgsImV4cCI6MTYzNzg2MzI0OH0.T30MrpWmFNVNOQ6\_ZVsTR5Or-MsSVAwTWtKsy\_cpMjo",

    "id": 1

}

Đây là refreshToken: dùng cho logout, được lưu trong cookie

refreshToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDkzNTA0NDU4NTM1MzM4MzkyMTUiLCJpYXQiOjE2Mzc4NDkzNDQsImV4cCI6MTY0MDQ0MTM0NH0.fbJmVEhS736WIB3Y2M4llkors5tvmYEdfaQ8\_H5KOaQ; Path=/; HttpOnly; Expires=Sat, 25 Dec 2021 14:09:04 GMT;

2. Refresh Token

- Mục đích: lấy access token mới khi access token hiện tại hết hạn

- Phương thức: POST

- URL: localhost:3001/auth/refresh-token

- Tham số: lấy tham số refreshToken từ cookie, không cần truyền thêm vào body

- Dữ liệu trả về:

{

    "success": **true**,

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDQwMTkwNzA4MTIwNjEwMDA0NTkiLCJpYXQiOjE2Mzc4NDU0MzUsImV4cCI6MTYzNzg2MzQzNX0.VbcguYZ3PFrv4tAZ-jrRD2B4yckyKx7vM7hvVvwLSY8"

}

3. Logout

- Mục đích: đăng xuất tài khoản

- Phương thức: POST

- URL: localhost:3001/auth/logout

- Tham số: lấy tham số refreshToken từ cookie, không cần truyền thêm vào body

- Dữ liệu trả về:

{

    "success": **true //đăng xuất tài khoản thành công**

}

4. Get user theo user id

- Mục đích: lấy thông tin user theo user id

- Phương thức: GET

- URL: localhost:3001/user/id/ + userId, ví dụ localhost:3001/user/id/1

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 1,

        "userName": "admin01",

        "password": "$2b$10$1KomzNqcn1nTTo4sY4s7euzNX8t1EVBVOht/psv91sd1NvrL1D6ZG",

        "email": "vuquanghuy21081999@gmail.com",

        "avatarLink": "./public/images/default\_avatar.jpg",

        "gender": 1,

        "facebookLink": "",

        "githubLink": "",

        "location": "Ha Noi",

        "description": "Admin account",

        "date": "2021-11-25T08:04:44.000Z",

        "role": 2,

        "googleId": "104019070812061000459"

    }

}

5. Get user theo username

- Mục đích: lấy thông tin user từ username(validate lúc đăng ký,…)

- Phương thức: GET

- URL: localhost:3001/user/username/ + username (ví dụ: localhost:3001/user/username/admin01)

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 1,

        "userName": "admin01",

        "password": "$2b$10$1KomzNqcn1nTTo4sY4s7euzNX8t1EVBVOht/psv91sd1NvrL1D6ZG",

        "email": "vuquanghuy21081999@gmail.com",

        "avatarLink": "./public/images/default\_avatar.jpg",

        "gender": 1,

        "facebookLink": "",

        "githubLink": "",

        "location": "Ha Noi",

        "description": "Admin account",

        "date": "2021-11-25T08:04:44.000Z",

        "role": 1, //1 là admin còn 0 là user

        "googleId": "104019070812061000459"

    }

}

6. Get user theo email

- Mục đích: lấy thông tin user theo email (validate lúc đăng ký)

- Phương thức: GET

- URL: localhost:3001/user/email/ + user email

(ví dụ localhost:3001/user/email/vuquanghuy21081999@gmail.com)

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 1,

        "userName": "admin01",

        "password": "$2b$10$1KomzNqcn1nTTo4sY4s7euzNX8t1EVBVOht/psv91sd1NvrL1D6ZG",

        "email": "vuquanghuy21081999@gmail.com",

        "avatarLink": "./public/images/default\_avatar.jpg",

        "gender": 1,

        "facebookLink": "",

        "githubLink": "",

        "location": "Ha Noi",

        "description": "Admin account",

        "date": "2021-11-25T08:04:44.000Z",

        "role": 1,

        "googleId": "104019070812061000459"

    }

}

6. Upload avatar

- Mục đích: upload ảnh làm avatar cho người dùng

- Phương thức: POST

- URL: http://localhost:3001/user/upload-avatar

- Tham số: gửi trong body

|  |  |
| --- | --- |
| Key | Value |
| avatar | File ảnh (jpeg|jpg|png) |

- Dữ liệu trả về:

{

    "success": **true**,

    "result": "public\\images\\avatar-1637846554629.jpg" //đường dẫn sử dụng trong avatar link lúc đăng ký

}

7.Send OTP khi quên mật khẩu

- Mục đích: gửi OTP cho người dùng qua email khi quên mật khẩu

- Phương thức: POST

- URL: localhost:3001/auth/forget-password/send-otp

- Tham số trong body

{

    "email": [vuquanghuy21081999@gmail.com](mailto:vuquanghuy21081999@gmail.com) //email của tài khoản cần lấy

}

- Dữ liệu trả về:

{

    "success": **true**,

    "otpToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDQwMTkwNzA4MTIwNjEwMDA0NTkiLCJpYXQiOjE2Mzc4NDczNzIsImV4cCI6MTYzNzg0NzU1Mn0.14EvCNYIUBSC3qKn2vpUGd4Sqhy0wtirKlPFIn\_PKag"

// otpToken được sử dụng để xác định xem là mã OTP đã hết hạn hay chưa (thời gian hiệu lực là 3 phút

}

8. Check OTP

- Mục đích: check OTP của người dùng xem có hợp lệ (đúng và còn hạn hay không)

- Phương thức: POST

- URL: localhost:3001/auth/forget-password/check-otp

- Tham số gửi lên trong body

{

    "otp":174921, //mã otp nhận được từ email

    "otpToken":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDQwMTkwNzA4MTIwNjEwMDA0NTkiLCJpYXQiOjE2Mzc4NDczNzIsImV4cCI6MTYzNzg0NzU1Mn0.14EvCNYIUBSC3qKn2vpUGd4Sqhy0wtirKlPFIn\_PKag"

//token xác định xem mã đó còn hạn hay không

}

- Dữ liệu trả về

{

    "success": **true**,

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDQwMTkwNzA4MTIwNjEwMDA0NTkiLCJpYXQiOjE2Mzc4NDc3NjgsImV4cCI6MTYzNzg2NTc2OH0.JSF6XLwuwliuisFlxSbZ05glK76VhV7dOKw2Kpc5SpQ"

// access token của user tương ứng, dùng access token này để tiến hành đổi mật khẩu

}

9. Forget Password

- Mục đích: tạo mật khẩu mới cho tài khoản khi quên mật khẩu và tiến hành set lại thông qua OTP

- Phương thức: POST

- URL: localhost:3001/auth/forget-password/reset-password

- Tham số gửi lên trong body:

{

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDQwMTkwNzA4MTIwNjEwMDA0NTkiLCJpYXQiOjE2Mzc4NDc3NjgsImV4cCI6MTYzNzg2NTc2OH0.JSF6XLwuwliuisFlxSbZ05glK76VhV7dOKw2Kpc5SpQ",

    "newPassword": "huython" // password mới

}

10. Login bằng tài khoản google

- Mục đích: đăng nhập vào hệ thống bằng tài khoản google (với user đã liên kết tài khoản hệ thống vs tài khoản google)

- Phương thức: POST

- URL: localhost:3001/auth/login-with-google

- Dữ liệu gửi trong body

"idToken": "eyJhbGciOiJSUzI1NiIsImtpZCI6ImQ0ZTA2Y2ViMjJiMDFiZTU2YzIxM2M5ODU0MGFiNTYzYmZmNWE1OGMiLCJ0eXAiOiJKV1QifQ..DeNcf5cLUJ0IML2VraGzGqnQ2SoOEKDb2ZWCwkZQsfwhjuahR8tIPIyJcsbGAtS2mVkCwG5AP2aUGACC8jpiZSD\_Au4M\_UrrkODRD8RM2eTk9z3sgeMrePSw57ZcIukjz5PGgEzf\_9\_1M52Qa8Bnu3jhLQ7a6f9orgz-ybjA-JfxO2salDUXstlaV7JPloqonFrA5Nr91BShfV-1njoF71BHF7WHQGTF\_vdGS1J6yUxjXQESf\_5kfhI\_np3UXwFROTJce-OSvZ6JP0v-eA02NOwiSbMfytUu8isk0L-5K-DXyWjOVOvTtzaSz78PR3PJQ2RcfCqSrNjdWtzSJg\_Mdw"

//đây là token lấy được khi frontend đăng nhập thông qua google, lấy từ trường tokenId

- Dữ liệu trả về

+ Trường hợp đăng nhập thành công

{

    "success": **true**, //thành công khi lấy dữ liệu từ google

    "exist": **true**,//bằng true khi tài khoản hệ thống đã liên kết với tài khoản google hoặc tài khoản hệ thống có email giống vs tài khoản google (khi giống email thì tiến hành liên kết tài khoản luôn)

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDkzNTA0NDU4NTM1MzM4MzkyMTUiLCJpYXQiOjE2Mzc4NDg1MDAsImV4cCI6MTYzNzg2NjUwMH0.9B5fgy3VOvb5XNm\_-Dl5bXnpLTyWUfcs0zpyiWb2zBs",

//access token

    "message": "Login successfully",

//dưới đây là các tham số của google trả về, cái nào quan trọng thì mới có comment

    "iss": "accounts.google.com",

    "azp": "846925863530-akeqgusm19k4dgicm7ncbl1jlsdi48m2.apps.googleusercontent.com",

    "aud": "846925863530-akeqgusm19k4dgicm7ncbl1jlsdi48m2.apps.googleusercontent.com",

    "sub": "109350445853533839215", //google ID, unique cho mỗi tài khoản

    "email": "vuquanghuy21081999@gmail.com",

    "email\_verified": **true**,

    "at\_hash": "vNE93MFgAyxy90FNtuSPcQ",

    "name": "Huy Thộn", //tên tài khoản

    "picture": "https://lh3.googleusercontent.com/a/AATXAJxXQRHOPuvgYIku\_DKHAY5iSSijTKuhIa9BQMvI=s96-c", //link avatar google, có thể dùng làm ảnh đại diện khi đăng ký

    "given\_name": "Huy",

    "family\_name": "Thộn",

    "locale": "vi",

    "iat": 1637848301,

    "exp": 1637851901,

    "jti": "3e5ee3784e6a1755b7ae05ed800b74d07498382d"

}

Đây là refreshToken: dùng cho logout, được lưu trong cookie

refreshToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDkzNTA0NDU4NTM1MzM4MzkyMTUiLCJpYXQiOjE2Mzc4NDkzNDQsImV4cCI6MTY0MDQ0MTM0NH0.fbJmVEhS736WIB3Y2M4llkors5tvmYEdfaQ8\_H5KOaQ; Path=/; HttpOnly; Expires=Sat, 25 Dec 2021 14:09:04 GMT;

+ Trường hợp đăng nhập không thành công

{

    "success": **true**,

    "exist": **false**,

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6Imh1eXZxLnNvaWN0Lmh1c3RAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDEyNTE2NzIxNTgxODUzMzk0ODYiLCJpYXQiOjE2MzgwMTk5NTIsImV4cCI6MTYzODAzNzk1Mn0.uzRGSmQg22YsAevxlNMhpUNd6gqwU-QE0au80N-Xr70",

//dùng access token này để mà dùng API đăng ký tài khoản (xem mục 13)

    "message": "User does not exist, please sign up",

    "iss": "accounts.google.com",

    "azp": "846925863530-akeqgusm19k4dgicm7ncbl1jlsdi48m2.apps.googleusercontent.com",

    "aud": "846925863530-akeqgusm19k4dgicm7ncbl1jlsdi48m2.apps.googleusercontent.com",

    "sub": "101251672158185339486",

    "email": "huyvq.soict.hust@gmail.com",

    "email\_verified": **true**,

    "at\_hash": "1p151Bi9WwbJC9MvUa43lw",

    "name": "Quang Huy Vũ",

    "picture": "https://lh3.googleusercontent.com/a-/AOh14GgplE35dg2fjRtXdtazgzPMjZAnAurd4Ve\_DEh5=s96-c",

    "given\_name": "Quang Huy",

    "family\_name": "Vũ",

    "locale": "vi",

    "iat": 1638019927,

    "exp": 1638023527,

    "jti": "a1350bfbee81621d1de6fb0e619aa43fa3587ca5"

}

- Thông tin về luồng hoạt động

+ Nếu tài khoản hệ thống đã liên kết với tài khoản google thì tiến hành đăng nhập luôn

+ Nếu tài khoản hệ thống chưa liên kết với tài khoản google, nhưng có email giống với google email, thì tiến hành liên kết tài khoản hệ thống với google, sau đó đăng nhập luôn

+ Nếu không tồn tại, thì trả cho người dùng 1 access token, sau đó gửi token này kèm thông tin tài khoản khi đăng ký (xem mục 13)

11. Send OTP khi đăng ký bằng email

- Mục đích: gửi mã OTP về email cho người dùng khi họ tiến hành đăng ký tài khoản hệ thống bằng email

- Phương thức: POST

- URL: localhost:3001/auth/sign-up/send-otp

- Dữ liệu gửi lên trong body

{

    "email": [ahnheeyeon2108@gmail.com](mailto:ahnheeyeon2108@gmail.com) //email đăng ký

}

- Dữ liệu trả về

{

    "success": **true**,

    "otpToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImFobmhlZXllb24yMTA4QGdtYWlsLmNvbSIsImlhdCI6MTYzODAyMDQyMywiZXhwIjoxNjM4MDIwNjAzfQ.w3VpPQt0BD3MYJZBY0rt08Ulo3RIQUofEUScSBbhkFU" //token để xác định xem otp có hợp lệ (đúng và còn hạn không)

}

12. Check OTP khi đăng ký bằng email

- Mục đích: kiểm trả mã OTP mà người dùng nhập vào xem nó còn có hợp lệ hay không (đúng và còn hạn)

- Phương thức: POST

- URL: localhost:3001/auth/sign-up/check-otp

- Dữ liệu gửi lên trong body

{

    "otp":"629431", //mã OTP người dùng lấy từ email và nhập vào

    "otpToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImFobmhlZXllb24yMTA4QGdtYWlsLmNvbSIsImlhdCI6MTYzODAyMDU2MCwiZXhwIjoxNjM4MDIwNzQwfQ.vy4OrT\_M-lwLang7J6Dkpf5uHM5Pc25JCS7wGe6KIkw" //OTP token được gửi từ bước sent otp để dăng ký trước đó

}

- Dữ liệu trả về

{

    "success": **true**,

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImFobmhlZXllb24yMTA4QGdtYWlsLmNvbSIsImlhdCI6MTYzODAyMDU5NiwiZXhwIjoxNjM4MDM4NTk2fQ.\_n5VcUfM1oMjc\_\_fRU04uf4Ym9nfTZdzTet\_ZdaWsRI"

// access token, được sinh ra từ email (và googleId khi đăng ký bằng google) dùng access token này để tiến hành đăng ký (sẽ mô tả sau đây)

}

13. Sign up

- Mục đích: gửi thông tin đăng ký của người dùng lên hệ thống để hoàn tất quá trình đăng ký cuối cùng

- Phương thức: POST

- URL: localhost:3001/auth/sign-up

- Dữ liệu gửi lên trong body

{

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6Imh1eXZxLnNvaWN0Lmh1c3RAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDEyNTE2NzIxNTgxODUzMzk0ODYiLCJpYXQiOjE2MzgwMTk5NTIsImV4cCI6MTYzODAzNzk1Mn0.uzRGSmQg22YsAevxlNMhpUNd6gqwU-QE0au80N-Xr70",

//access token lấy được sau bước check OTP đăng ký (mục 12) hoặc bước login với google (mục 10)

    "userName": "admin10",

    "password": "huython",

    "email": "huyvq.soict.hust@gmail.com",

//phải trùng với email đã dùng để gửi OTP (mục 11, 12) hoặc login google không thành công (mục 8) hoặc đăng ký bằng tài khoản google (mục 14 sau đây)

    "avatarLink": "public\\images\\avatar-1637672436349.jpg",

    "gender": **true**,

    "facebookLink": "https://www.facebook.com/huython.2108",

    "githubLink": "",

    "location": "Ha Noi",

    "description": "Admin sieu cap vjp pro",

    "role": 1,

    "googleID":""// phải có khi login google không thành công rồi chuyển sang đây (mục 8) hoặc đăng ký bằng tài khoản google (mục 14 sau đây)

}

- Dữ liệu trả về:

{

    "success": **true**,

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6WzEyXSwiZW1haWwiOiJodXl2cS5zb2ljdC5odXN0QGdtYWlsLmNvbSIsImlhdCI6MTYzODAyMTEwOCwiZXhwIjoxNjM4MDM5MTA4fQ.AjzskZYl3Lny4j2tt7-x6kS93HsIGkCVyggdRYA7kEs"

}

Refresh token gửi về trong cookie

refreshToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6WzEyXSwiZW1haWwiOiJodXl2cS5zb2ljdC5odXN0QGdtYWlsLmNvbSIsImlhdCI6MTYzODAyMTEwOCwiZXhwIjoxNjQwNjEzMTA4fQ.48bykvEzJTcFM1i2Ltav6NqaJsK\_z7suNMt7T4xfcOc; Path=/; HttpOnly; Expires=Mon, 27 Dec 2021 13:51:48 GMT;

14. Sign up bằng tài khoản google

//Nói chung cái này giống với Login bằng tài khoản google

- Mục đích: đăng ký bằng tài khoản google, nếu đã có tài khoản hệ thống liên kết với tài khoản google này thì tiến hành đăng nhập luôn, nếu email tài khoản trùng với email google thì liên kết tài khoản hệ thống với tài khoản google và đăng nhập luôn. Nếu không có thì trả về access token và dùng API sign up để nhập thông tin (mục 13) và đăng ký

- Phương thức: POST

- URL: localhost:3001/auth/sign-up-with-google

- Dữ liệu gửi lên trong body

{

    "idToken": "eyJhbGciOiJSUzI1NiIsImtpZCI6ImQ0ZTA2Y2ViMjJiMDFiZTU2YzIxM2M5ODU0MGFiNTYzYmZmNWE1OGMiLCJ0eXAiOiJKV1QifQ..Lrx-y7D\_iWVxOkXOSkTvpfiD0iYXT9dMJSnMK9wpyDLZE4d3PXHdGJMb2WNIcQFOYdATXuZhfC098DhG1h4vUauePT-QWEo3SDeztdW7TpbYDbh5dlzOcOXzZw92IBd7zcK\_VXjmWGtpVMgpOczNzqBoupL5V\_ie\_h\_PtYknFYeEW5V4rU5iaKLHxbgCaoG87YlcxfP13JputDGbGaYKyITMIzEXY9OlT-2h0qyJYi7CiWirfz4npXSWBqqcdKucu7n3Ed9tvYEkItjtSa4lEKt1p9LgtBo7xtpZpZBKJixjHRqfOyLvPeGHlaobTaplrncDOCk5zMMdTOlP2zufnw"

}

//Token Id do google gửi về

- Kết quả trả về:

{

    "success": **true**,

    "exist": **false**,

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImRvYW50aGlldGtlaHR0dDIwMjExQGdtYWlsLmNvbSIsImdvb2dsZUlkIjoiMTEyMzM4MDExOTY1NTQ5NDc0MTY2IiwiaWF0IjoxNjM4MDIzNDA0LCJleHAiOjE2MzgwNDE0MDR9.YVxHqAZWGmQbOssemS0iv1uERZKmJIZt8FIp0XfFKDA",

    "message": "Can create account with this google account, please sign up with this access token",

    "iss": "accounts.google.com",

    "azp": "846925863530-akeqgusm19k4dgicm7ncbl1jlsdi48m2.apps.googleusercontent.com",

    "aud": "846925863530-akeqgusm19k4dgicm7ncbl1jlsdi48m2.apps.googleusercontent.com",

    "sub": "112338011965549474166",

    "email": "doanthietkehttt20211@gmail.com",

    "email\_verified": **true**,

    "at\_hash": "uhtmEUjE-2EqQ1q-3qiUDQ",

    "name": "Huy Vu",

    "picture": "https://lh3.googleusercontent.com/a/AATXAJxiqtk\_FUmkFbOLlGrh2kB4exT8cCrEudb6CfiE=s96-c",

    "given\_name": "Huy",

    "family\_name": "Vu",

    "locale": "en",

    "iat": 1638023344,

    "exp": 1638026944,

    "jti": "1af2d770c980917f38c1910e9ffaecd838fb3c94"

}

- Thông tin về luồng hoạt động

+ Nếu tài khoản hệ thống đã liên kết với tài khoản google thì tiến hành đăng nhập luôn

+ Nếu tài khoản hệ thống chưa liên kết với tài khoản google, nhưng có email giống với google email, thì tiến hành liên kết tài khoản hệ thống với google, sau đó đăng nhập luôn

+ Nếu không tồn tại, thì trả cho người dùng 1 access token, sau đó gửi token này kèm thông tin tài khoản khi dùng API sign up để đăng ký (xem mục 13)

\*\*\* Lưu ý: một số API có yêu cầu quyền user mới có thể truy cập được, tức là bạn phải gửi access token (trong API là token) lên. Nếu gặp lỗi 401 thì do không thêm access token vào request

15. Link with google account

- Mục đích: kết nối tài khoản hệ thống với tài khoản google

- Phương thức: POST

- URL: localhost:3001/user/link-with-google-account

- Dữ liệu gửi trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzODUzNTU2MSwiZXhwIjoxNjM4NTUzNTYxfQ.zi4qvzAS5RunTJ07-xsOKJXDNgwb5BvHLlTU0kC7ubY",

//access token của người dùng

    "googleId": 109350445853533839215

}

- Dữ liệu trả về:

{

    "success": **true //liên kết có thành công hay không**

}

16. Edit profile

- Mục đích: thay đổi thông tin profile của tài khoản (trừ password và email)

- Phương thức: POST

- URL: localhost:3001/user/edit-profile

- Dữ liệu gửi trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzODUzNTU2MSwiZXhwIjoxNjM4NTUzNTYxfQ.zi4qvzAS5RunTJ07-xsOKJXDNgwb5BvHLlTU0kC7ubY", //access token

    "userName": "vuquanghuy2108",

    "gender": **false**,

    "facebookLink": "https://www.facebook.com/huython.2108",

    "githubLink": "https://github.com/huythonhd1999",

    "location": "Hải Dương",

    "description": "Đổi 15h ngày 30/11/2021"

}

- Dữ liệu trả về:

{

    "success": **true**,

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDkzNTA0NDU4NTM1MzM4MzAwMDAiLCJpYXQiOjE2Mzg1MzU3NTksImV4cCI6MTYzODU1Mzc1OX0.vx03HWfp-5krLVVo\_rbC6j-KxQg1gTlRnMxrlLmnKiE",

    "user": {

        "Id": 1,

        "userName": "vuquanghuy2108",

        "password": "$2b$10$Mo.6XHjqWg32QW.wRQ1zju5.nxybb89bcdEsKvRwIJwl8nSZxAEzi",

        "email": "vuquanghuy21081999@gmail.com",

        "avatarLink": "./public/images/default\_avatar.jpg",

        "gender": 0,

        "facebookLink": "https://www.facebook.com/huython.2108",

        "githubLink": "https://github.com/huythonhd1999",

        "location": "Hải Dương",

        "description": "Đổi 15h ngày 30/11/2021",

        "date": "2021-12-02T13:25:44.000Z",

        "role": 1,

        "googleId": "109350445853533830000"

    },

    "message": "Change profile successful"

}

Refresh token trong cookie:

refreshToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJhaG5oZWV5ZW9uMjEwOEBnbWFpbC5jb20iLCJnb29nbGVJZCI6IjEwOTM1MDQ0NTg1MzUzMzgzMDAwMCIsImlhdCI6MTYzODUzNjMyMCwiZXhwIjoxNjQxMTI4MzIwfQ.strAG3YsqHKnNwhzUZkG8flEK\_r7QgjJd9Ei1whrQKg; Path=/; HttpOnly; Expires=Sun, 02 Jan 2022 12:58:40 GMT;

17. Sent otp để dổi email

- Mục đích: gửi OTP cho email mới khi người dùng muốn đổi email cho tài khoản

- Phương thức: POST

- URL: localhost:3001/user/edit-user/edit-email/sent-otp

- Dữ liệu gửi trên body  
{

    "email": "ahnheeyeon2108@gmail.com", //email mới người dùng nhập vào

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDkzNTA0NDU4NTM1MzM4MzAwMDAiLCJpYXQiOjE2Mzg1MzU3NTksImV4cCI6MTYzODU1Mzc1OX0.vx03HWfp-5krLVVo\_rbC6j-KxQg1gTlRnMxrlLmnKiE" //access token của tài khoản hiện tại

}

- Dữ liệu trả về

{

    "success": **true**,

    "otpToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJhaG5oZWV5ZW9uMjEwOEBnbWFpbC5jb20iLCJpYXQiOjE2Mzg1MzU5MjYsImV4cCI6MTYzODUzNjEwNn0.u4jDfoTp2U5g0Qo0bIxlCRz64RaNgFfCeKOrK0XLZzM"

}

18. Check otp và set email

- Mục đích: kiểm tra mã OTP có hợp lệ hay không, nếu có thì set lại email của tài khoản

- Phương thức: POST

- URL: localhost:3001/user/edit-user/edit-email/check-otp

- Dữ liệu gửi lên

{

    "otp":548981,

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDkzNTA0NDU4NTM1MzM4MzAwMDAiLCJpYXQiOjE2Mzg1MzU3NTksImV4cCI6MTYzODU1Mzc1OX0.vx03HWfp-5krLVVo\_rbC6j-KxQg1gTlRnMxrlLmnKiE", //access token của tài khoản

    "otpToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwiZW1haWwiOiJhaG5oZWV5ZW9uMjEwOEBnbWFpbC5jb20iLCJpYXQiOjE2Mzg1MzYwNjUsImV4cCI6MTYzODUzNjI0NX0.OGRCy-dMshuZ8DPqPcFN7bPKCV0-nRrM6oqgU5d0e\_o" //otp token được lấy từ bước trước

}

- Dữ liệu trả về

{

    "success": **true**,

    "user": {

        "Id": 1,

        "userName": "vuquanghuy2108",

        "password": "$2b$10$Mo.6XHjqWg32QW.wRQ1zju5.nxybb89bcdEsKvRwIJwl8nSZxAEzi",

        "email": "ahnheeyeon2108@gmail.com",

        "avatarLink": "./public/images/default\_avatar.jpg",

        "gender": 0,

        "facebookLink": "https://www.facebook.com/huython.2108",

        "githubLink": "https://github.com/huythonhd1999",

        "location": "Hải Dương",

        "description": "Đổi 15h ngày 30/11/2021",

        "date": "2021-12-02T13:25:44.000Z",

        "role": 1,

        "googleId": "109350445853533830000"

    },

    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJhaG5oZWV5ZW9uMjEwOEBnbWFpbC5jb20iLCJnb29nbGVJZCI6IjEwOTM1MDQ0NTg1MzUzMzgzMDAwMCIsImlhdCI6MTYzODUzNjEyNSwiZXhwIjoxNjM4NTU0MTI1fQ.wT2sfGhI9Adp0jQNRu11Ve50hW5\_pWLezxdb7FOf7X0"

}

Refresh token gửi trong cookie

refreshToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDkzNTA0NDU4NTM1MzM4MzAwMDAiLCJpYXQiOjE2Mzg1MzY0ODQsImV4cCI6MTY0MTEyODQ4NH0.751srUmrxEai7NGyuQaDLLF6I7yNBaKpaMYj5sqpxDo; Path=/; HttpOnly; Expires=Sun, 02 Jan 2022 13:01:24 GMT;

19. Change password

- Mục đích: thay đổi mật khẩu khi biết mật khẩu cũ

- Phương thức: POST

- URL: localhost:3001/user/edit-user/change-password

- Dữ liệu gửi lên trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDkzNTA0NDU4NTM1MzM4MzAwMDAiLCJpYXQiOjE2Mzg1MzY0ODQsImV4cCI6MTYzODU1NDQ4NH0.Vk2uqiNZvmuHJJiFJqdLRf-YfmY9-Zy26pMUbRIza24",

    "oldPassword":"a12345678",

    "newPassword": "vuquanghuy2108"

}

- Dữ liệu trả về:

{

    "success": **true**,

    "message": "Change password successful"

}

20. Get tag by tag ID

- Mục đích: Lấy thông tin tag theo tag ID

- Phương thức: GET

- URL: localhost:3001/tag/id/{tagID}, ví dụ localhost:3001/tag/id/2

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 2,

        "userId": 1,

        "tagName": "c#",

        "tagDetail": "C# (pronounced 'see sharp') is a high level, statically typed, multi-paradigm programming language developed by Microsoft. C# code usually targets Microsoft's .NET family of tools and run-times, which include .NET, .NET Framework and Xamarin among others. Use this tag for questions about code written in C# or about C#'s formal specification.",

        "date": "2021-12-02T13:25:44.000Z" //ngày tạo tag theo giờ GMT + 0

    }

}

21. Get tag list theo page

- Mục đích: lấy danh sách tag list, có phân trang và có thể sắp xếp

- Phương thức: GET

- URL:

localhost:3001/tag/list?page={page}&perPage={perPage}&orderBy={orderColumn}&orderType={orderType}

ví dụ: localhost:3001/tag/list?page=1&perPage=5&orderBy=postNum&orderType=desc

- Dữ liệu gửi lên trong params (đã trong URL rồi)

+ page: page cần lấy

+ perPage: số lượng mỗi tag trong 1 page

+ orderBy: sắp xếp theo cột nào (mặc định nếu không truyền gì là Id của tag), các cột dùng được là

. date: thời gian tạo tag

. postNum: số lượng post có chứa tag này

. tagName: tên tag

. …

+ orderType: loại sắp xếp, chọn ‘asc’ hoặc ‘desc’ (mặc định là asc)

- Dữ liệu trả về

{

    "success": **true**,

    "result": {

        "data": [

            {

                "Id": 1,

                "userId": 1,

                "tagName": "Java",

                "tagDetail": "Java is a high-level object oriented programming language. Use this tag when you're having problems using or understanding the language itself. This tag is frequently used alongside other tags for libraries and/or frameworks used by Java developers.",

                "date": "2021-12-14T08:24:43.000Z",

                "postNum": 3

            },

            {

                "Id": 2,

                "userId": 1,

                "tagName": "c#",

                "tagDetail": "C# (pronounced 'see sharp') is a high level, statically typed, multi-paradigm programming language developed by Microsoft. C# code usually targets Microsoft's .NET family of tools and run-times, which include .NET, .NET Framework and Xamarin among others. Use this tag for questions about code written in C# or about C#'s formal specification.",

                "date": "2021-12-14T08:24:43.000Z",

                "postNum": 2

            }

        ],

        "pagination": {

            "total": 6,

            "lastPage": 3,

            "perPage": 2,

            "currentPage": 1,

            "from": 0,

            "to": 2

        }

    }

}

22. Lấy list tag của người dùng có phân trang

- Mục đích: lấy danh sách list tag của người dùng tạo, có thể sắp xếp

- Phương thức: POST

- URL: localhost:3001/tag/list?userId={userID}&page={page}&perPage={perPage}&orderBy={orderColumn}&orderType={orderType}

Ví dụ: localhost:3001/tag/list?userId=1&page=1&perPage=2&orderBy=date&orderType=desc

- Thuộc tính trong params (đã có trên URL)

+ userID: id của người dùng cần lấy (tự truyền id của mình cũng được)

+ page: page cần lấy

+ perPage: số lượng mỗi tag trong 1 page

+ orderBy: sắp xếp theo cột nào (mặc định nếu không truyền gì là Id của tag), các cột dùng được là

. date: thời gian tạo tag

. postNum: số lượng post có chứa tag này

. tagName: tên tag

. …

+ orderType: loại sắp xếp, chọn ‘asc’ hoặc ‘desc’ (mặc định là asc)

- Dữ liệu trả về

{

    "success": **true**,

    "result": {

        "data": [

            {

                "Id": 6,

                "userId": 1,

                "tagName": "Them thử tag",

                "tagDetail": "Test",

                "date": "2021-12-14T08:26:08.000Z",

                "postNum": 0

            },

            {

                "Id": 1,

                "userId": 1,

                "tagName": "Java",

                "tagDetail": "Java is a high-level object oriented programming language. Use this tag when you're having problems using or understanding the language itself. This tag is frequently used alongside other tags for libraries and/or frameworks used by Java developers.",

                "date": "2021-12-14T08:24:43.000Z",

                "postNum": 3

            }

        ],

        "pagination": {

            "total": 6,

            "lastPage": 3,

            "perPage": 2,

            "currentPage": 1,

            "from": 0,

            "to": 2

        }

    }

}

23. Search tag

- Mục đích: tìm kiếm tag theo tên tag, có phân trang và sắp xếp

- Phương thức: GET

- URL:

localhost:3001/tag/search?query={query}&page={page}&perPage={perPage}&orderBy=orderBy}&orderType={sortType}

ví dụ localhost:3001/tag/search?query=t&page=1&perPage=2&orderBy=date&orderType=desc

- Dữ liệu truyền trong params (đã có trong URL):

+ query: tên tag cần search

+ page: page cần lấy

+ perPage: số lượng mỗi tag trong 1 page

+ orderBy: sắp xếp theo cột nào (mặc định nếu không truyền gì là Id của tag), các cột dùng được là

. date: thời gian tạo tag

. postNum: số lượng post có chứa tag này

. tagName: tên tag

. …

+ orderType: loại sắp xếp, chọn ‘asc’ hoặc ‘desc’ (mặc định là asc)

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "data": [

            {

                "Id": 6,

                "userId": 1,

                "tagName": "Them thử tag",

                "tagDetail": "Test",

                "date": "2021-12-14T08:26:08.000Z",

                "postNum": 0

            },

            {

                "Id": 3,

                "userId": 1,

                "tagName": "javascript",

                "tagDetail": "For questions regarding programming in ECMAScript (JavaScript/JS) and its various dialects/implementations (excluding ActionScript).",

                "date": "2021-12-14T08:24:43.000Z",

                "postNum": 1

            }

        ],

        "pagination": {

            "total": 3,

            "lastPage": 2,

            "perPage": 2,

            "currentPage": 1,

            "from": 0,

            "to": 2

        }

    }

}

24. Create tag

- Mục đích: tạo tag mới

- Phương thức: POST

- URL: localhost:3001/tag/create-tag

- Dữ liệu gửi lên trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTQ4NzU5NywiZXhwIjoxNjM5NTA1NTk3fQ.8f589Lb28pmbXXHs7mvtCSebL7WtdzjknsrxJeUVl7U",

    "tagName": "Them thử tag 1", //tag name unique

    "tagDetail": "Test thử edit tag" //cung unique

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 9,

        "userId": 1,

        "tagName": "Them thử tag 1",

        "tagDetail": "Test thử edit tag",

        "date": "2021-12-14T13:13:52.000Z",

        "postNum": 0

    }

}

25. Sửa tag

- Mục đích: sửa tag do người dùng tạo ra (hoặc là admin thì auto sửa được)

- Phương thức: POST

- URL: localhost:3001/tag/edit-tag

- Dữ liệu gửi lên trong body:

- Dữ liệu trả về

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTQ4NzU5NywiZXhwIjoxNjM5NTA1NTk3fQ.8f589Lb28pmbXXHs7mvtCSebL7WtdzjknsrxJeUVl7U",

    "tagId": 1, //tagId cần edit

    "tagName": "test thử edit tag 14/12",

    "tagDetail": "test lúc 10h ngày 14/12/2021"

}

- Dữ liệu trả về

{

    "success": **true**,

    "result": {

        "Id": 1,

        "userId": 1,

        "tagName": "test thử edit tag 14/12",

        "tagDetail": "test lúc 10h ngày 14/12/2021",

        "date": "2021-12-14T08:24:43.000Z",

        "postNum": 3

    }

}

26. Xoá tag

- Mục đích: xoá tag do người dùng tạo ra (hoặc là admin thì auto xoá được)

- Phương thức: POST

- URL: localhost:3001/tag/delete-tag

- Dữ liệu gửi lên trong body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOiIxMDkzNTA0NDU4NTM1MzM4MzAwMDAiLCJpYXQiOjE2Mzg1MzY0ODQsImV4cCI6MTYzODU1NDQ4NH0.Vk2uqiNZvmuHJJiFJqdLRf-YfmY9-Zy26pMUbRIza24",

    "tagId": 11 //tag id cần xoá

}

- Dữ liệu trả về

{

    "success": **true**,

    "result": 1 //số bản ghi tác động

}

27. Get avatar

- Mục đichs: lấy về avatar của người dùng theo id (dùng để hiện thị), nếu muốn lấy avatar mặc định lúc đăng ký thì set id = 0

- Phương thức POST

- URL: localhost:3001/user/get-avatar/id/{id},

ví dụ localhost:3001/user/get-avatar/id/3 (cần lấy avatar mặc định thì set bằng 0

- Dữ liệu trà về: file ảnh avatar của người dùng hoặc file avatar mặc định

Graphical user interface, text, application

Description automatically generated

28. Get post by Id

- Mục đích: lấy thông tin cơ bản của post theo id

- Phương thức: GET

- URL: localhost:3001/post/id/1

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 6,

        "userId": 1,

        "postName": "Artisan Storage:link doesnt work laravel 8",

        "postDetail": "Google polyline decoded is not what expected",

        "date": "2021-12-14T08:24:42.000Z",

        "rightAnswerID": **null**,

        "viewNum": 0,//số lượng view của post, cập nhật khi lấy list answer của post

"postUserName": "admin01", //tên user tạo post

        "numAnswer": 0, //số lượng answer của post

        "upVoteNum": 0, //số lượng upvote

        "downVoteNum": 0, //số lượng downvote

"likeNum": 1 //số user like post này (cho post vào danh sách favorite post)

    }

}

29. Get all post

- Mục đích: lấy danh sách tất cả post có phân trang và sắp xếp

- Phương thức: GET

- URL: localhost:3001/post/list?page={page}&perPage={perPage}&orderBy={orderBy}&orderType={orrderType},

ví dụ: localhost:3001/post/list?page=1&perPage=2&orderBy=viewNum&orderType=desc

- Dữ liệu gửi lên trong params (đã trong URL rồi)

+ page: page cần lấy

+ perPage: số lượng mỗi tag trong 1 page

+ orderBy: sắp xếp theo cột nào (mặc định nếu không truyền gì là Id của post), các cột dùng được là

. date: thời gian tạo post

. viewNum: số view của post này

. numAnswer: số answer của post

. upVoteNum: số upVote (dùng tạm cái này để sort theo vote, khi nào tôi làm được đoạn trừ hai cái up downvote đi để tính vote thì tôi sẽ nói)

. downVoteNum: số downVote

. …

+ orderType: loại sắp xếp, chọn ‘asc’ hoặc ‘desc’ (mặc định là asc)

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "data": [

            {

                "Id": 2,

                "userId": 1,

                "postName": "Google polyline decoded is not what expected",

                "postDetail": "Google polyline decoded is not what expected",

                "date": "2021-12-14T08:24:42.000Z",

                "rightAnswerID": **null**,

                "viewNum": 7,

                "numAnswer": 2,

                "upVoteNum": 0,

                "downVoteNum": 0,

"likeNum": 1,

"postUserName": "admin01"

            },

            {

                "Id": 1,

                "userId": 1,

                "postName": "How interlink external hyperlink/hotspot from WPF with internal hotspot on 3D object in Unity?",

                "postDetail": "Google polyline decoded is not what expected",

                "date": "2021-12-14T08:24:42.000Z",

                "rightAnswerID": **null**,

                "viewNum": 1,

                "numAnswer": 3,

                "upVoteNum": 0,

                "downVoteNum": 0,

"likeNum": 1,

"postUserName": "admin01"

            }

        ],

        "pagination": {

            "total": 8,

            "lastPage": 4,

            "perPage": 2,

            "currentPage": 1,

            "from": 0,

            "to": 2

        }

    }

}

30. Get list post của user có phân trang

- Mục đích: Lấy list post của người dùng có phân trang và sắp xếp

- Phương thức: get

- URL: localhost:3001/post/list?userId={userId}&page={page}&perPage={perPage}&orderBy={orderBy}&orderType={orrderType},

ví dụ: localhost:3001/post/list?userId=1&page=1&perPage=2&orderBy=viewNum&orderType=desc

- Dữ liệu gửi lên trong params (đã trong URL rồi)

+ userId: id của user cần lấy

+ page: page cần lấy

+ perPage: số lượng mỗi tag trong 1 page

+ orderBy: sắp xếp theo cột nào (mặc định nếu không truyền gì là Id của post), các cột dùng được là

. date: thời gian tạo post

. viewNum: số view của post này

. numAnswer: số answer của post

. upVoteNum: số upVote (dùng tạm cái này để sort theo vote, khi nào tôi làm được đoạn trừ hai cái up downvote đi để tính vote thì tôi sẽ nói)

. downVoteNum: số downVote

. …

+ orderType: loại sắp xếp, chọn ‘asc’ hoặc ‘desc’ (mặc định là asc)

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "data": [

            {

                "Id": 2,

                "userId": 1,

                "postName": "Google polyline decoded is not what expected",

                "postDetail": "Google polyline decoded is not what expected",

                "date": "2021-12-14T08:24:42.000Z",

                "rightAnswerID": **null**,

                "viewNum": 7,

                "numAnswer": 2,

                "upVoteNum": 0,

                "downVoteNum": 0,

"likeNum": 1,

"postUserName": "admin01"

            },

            {

                "Id": 1,

                "userId": 1,

                "postName": "How interlink external hyperlink/hotspot from WPF with internal hotspot on 3D object in Unity?",

                "postDetail": "Google polyline decoded is not what expected",

                "date": "2021-12-14T08:24:42.000Z",

                "rightAnswerID": **null**,

                "viewNum": 1,

                "numAnswer": 3,

                "upVoteNum": 0,

                "downVoteNum": 0,

"likeNum": 1,

"postUserName": "admin01"

            }

        ],

        "pagination": {

            "total": 8,

            "lastPage": 4,

            "perPage": 2,

            "currentPage": 1,

            "from": 0,

            "to": 2

        }

    }

}

31. Search post theo tên

- Mục đích: tìm kiếm post theo tên, có phân trang và sắp xếp

- Phương thức: GET

- URL:

localhost:3001/post/search?query={query}}&page={page}&perPage={perPage}&orderBy={orderBy}&orderType={orrderType},

ví dụ localhost:3001/post/search?query=how&page=1&perPage=5&orderBy=upVoteNum&orderType=desc

- Dữ liệu gửi lên trong params (đã trong URL rồi)

+ query: chuỗi cần search

+ page: page cần lấy

+ perPage: số lượng mỗi tag trong 1 page

+ orderBy: sắp xếp theo cột nào (mặc định nếu không truyền gì là Id của post), các cột dùng được là

. date: thời gian tạo post

. viewNum: số view của post này

. numAnswer: số answer của post

. upVoteNum: số upVote (dùng tạm cái này để sort theo vote, khi nào tôi làm được đoạn trừ hai cái up downvote đi để tính vote thì tôi sẽ nói)

. downVoteNum: số downVote

. …

+ orderType: loại sắp xếp, chọn ‘asc’ hoặc ‘desc’ (mặc định là asc)

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "data": [

            {

                "Id": 1,

                "userId": 1,

                "postName": "How interlink external hyperlink/hotspot from WPF with internal hotspot on 3D object in Unity?",

                "postDetail": "Google polyline decoded is not what expected",

                "date": "2021-12-14T06:13:21.000Z",

                "rightAnswerID": **null**,

                "viewNum": 3,

                "numAnswer": 2,

                "upVoteNum": 1,

                "downVoteNum": 1,

"likeNum": 1,

"postUserName": "admin01"

            },

            {

                "Id": 4,

                "userId": 1,

                "postName": "How to retrieve the respective value from the checkbox to display in the div based on search text??",

                "postDetail": "Google polyline decoded is not what expected",

                "date": "2021-12-14T06:13:21.000Z",

                "rightAnswerID": **null**,

                "viewNum": 0,

                "numAnswer": 2,

                "upVoteNum": 0,

                "downVoteNum": 0,

"likeNum": 1,

"postUserName": "admin01"

            },

            {

                "Id": 5,

                "userId": 1,

                "postName": "How to calculate the percentage of failed requests (HTTP status 401) using Splunk query?",

                "postDetail": "Google polyline decoded is not what expected",

                "date": "2021-12-14T06:13:21.000Z",

                "rightAnswerID": **null**,

                "viewNum": 0,

                "numAnswer": 2,

                "upVoteNum": 0,

                "downVoteNum": 0

            }

        ],

        "pagination": {

            "total": 3,

            "lastPage": 1,

            "perPage": 5,

            "currentPage": 1,

            "from": 0,

            "to": 3

        }

    }

}

32. Tạo post

- Mục đích: tạo post mới

- Phương thức: POST

- URL: localhost:3001/post/create-post

- Dữ liệu gửi lên trong body

{

    "token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTQ1ODgxMCwiZXhwIjoxNjM5NDc2ODEwfQ.soygiHN6iukmhWeoy8rEz8Y0UGv62LeQONavnFbZr7U",

    "postName":"Test tạo post ngày hôm nay",

    "postDetail": "test tao post ngay 14/12",

    "postTags": [1,2] //id các tag của bài viết, lấy bằng search tag theo name

}

- Dữ liệu trả về:

{

    "success": **true**,

    "message": [], //lỗi ko thêm được tag nào thì nó sẽ báo vào chỗ này

    "result": {

        "Id": 8,

        "userId": 1,

        "postName": "Test tạo post ngày hôm nay",

        "postDetail": "test tao post ngay 14/12",

        "date": "2021-12-14T08:25:06.000Z",

        "rightAnswerID": **null**,

        "viewNum": 0,

        "numAnswer": 0,

        "upVoteNum": 0,

        "downVoteNum": 0,

"likeNum": 1,

"postUserName": "admin01"

    }

}

33. Sửa post

- Mục đích: sửa các thông tin cơ bản của post, là admin hoặc chủ post

- Phương thức: POST

- URL: localhost:3001/post/edit-post

- Dữ liệu gửi lên trong body

{

    "token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTQ4ODY1MiwiZXhwIjoxNjM5NTA2NjUyfQ.OeCSUSV0IWTgiWKbd2DJSDjB61Y52aR26DRzAkTyps8",

    "postId":8, //id của post cần sửa

    "postName":"tesst edit post ngay 14/12",

    "postDetail": "test edit post ngay 14/12",

    "postTags": [1, 2] //danh sách tag id mới của post

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 8,

        "userId": 1,

        "postName": "tesst edit post ngay 14/12",

        "postDetail": "test edit post ngay 14/12",

        "date": "2021-12-14T08:25:06.000Z",

        "rightAnswerID": **null**,

        "viewNum": 0,

        "postUserName": "admin01",

        "numAnswer": 0,

        "upVoteNum": 0,

        "downVoteNum": 0,

        "likeNum": 0

    }

}

34. Xoá post

- Mục đích: xoá post theo postID, phải là admin hoặc chủ post

- Phương thức: POST

- URL: localhost:3001/post/delete-post

- Dữ liệu gửi lên trong body

{

    "token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzODg4NDgyMywiZXhwIjoxNjM4OTAyODIzfQ.Anmv5oepUCtDWD83ZARE7TVy3dDQ6DRbkjkV103Pvd8",

    "postId": 8

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": 1

}

35. Lấy danh sách các tag của mỗi post

- Mục đích: lấy thông tin chi tiết các tag của post theo postId

- Phương thức: GET

- URL: localhost:3001/post/tag?postId={postId}, ví dụ localhost:3001/post/tag?postId=9

- Dữ liệu trả về:

{

    "success": **true**,

    "result": [

        {

            "Id": 1,

            "userId": 1,

            "tagName": "test thử edit tag 14/12",

            "tagDetail": "test lúc 10h ngày 14/12/2021",

            "date": "2021-12-14T08:24:43.000Z",

            "postNum": 3

        },

        {

            "Id": 2,

            "userId": 1,

            "tagName": "c#",

            "tagDetail": "C# (pronounced 'see sharp') is a high level, statically typed, multi-paradigm programming language developed by Microsoft. C# code usually targets Microsoft's .NET family of tools and run-times, which include .NET, .NET Framework and Xamarin among others. Use this tag for questions about code written in C# or about C#'s formal specification.",

            "date": "2021-12-14T08:24:43.000Z",

            "postNum": 2

        }

    ]

}

36. Lấy số lượng upvote downvote của mỗi post

- Mục đích: lấy số lượng upvote và downvote của post

- Phương thức: GET

- URL: localhost:3001/post/vote-num?postId={postId}, ví dụ localhost:3001/post/vote-num?postId=4

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "upVote": 0, //số lượng upvote

        "downVote": 1 //số lượng downvote

    }

}

37. Lấy vote của user cho post theo postId

- Mục đích: lấy thông tin vote xem user đã vote cho post này hay chưa (theo postId)

- Phương thức: POST

- URL: localhost:3001/post/user/get-vote

- Dữ liệu gửi lên trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzODg4NDgyMywiZXhwIjoxNjM4OTAyODIzfQ.Anmv5oepUCtDWD83ZARE7TVy3dDQ6DRbkjkV103Pvd8",

    "postId": 4

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": [ //nếu không có thì trả về mảng rỗng

        {

            "postId": 4,

            "userId": 1,

            "voteType": 0, //0 là downvote, 1 là upvote

            "date": "2021-12-07T09:16:11.000Z"

        }

    ]

}

38. Tạo một vote mới cho post

- Mục đích: cho user tạo vote mới cho post (vote cũ bị xoá nếu có)

- Phương thức: POST

- URL: localhost:3001/post/user/create-vote

- Dữ liệu gửi lên trong body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzODg4NDgyMywiZXhwIjoxNjM4OTAyODIzfQ.Anmv5oepUCtDWD83ZARE7TVy3dDQ6DRbkjkV103Pvd8",

    "postId": 1, //postId của post muốn vote

    "voteType": **true**  //true là upvote, false là downvote

}

- Dữ liệu trả về

{

    "success": **true**,

    "result": [

        {

            "postId": 1,

            "userId": 1,

            "voteType": 1,

            "date": "2021-12-07T14:50:47.000Z"

        }

    ]

}

39. Xoá vote của post

- Mục đích: cho phép người dùng loại bỏ vote của post đó

- Phương thức: POST

- URL: localhost:3001/post/user/delete-vote

- Dữ liệu gửi lên trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzODg4NDgyMywiZXhwIjoxNjM4OTAyODIzfQ.Anmv5oepUCtDWD83ZARE7TVy3dDQ6DRbkjkV103Pvd8",

    "postId": 4

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": 1 //số lượng bản ghi can thiệp được, nếu bằng 0 tức là đã xoá rồi

}

40. Lấy answer theo id

- Mục đích: lấy thông tin answer theo id

- Phương thức: GET

- URL: localhost:3001/answer/id/{id}, ví dụ localhost:3001/answer/id/1

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 1,

        "userId": 1,

        "postId": 1,

        "answerDetail": " And there is only one entry in the NodeMachine table - the K1 wheel, which is suitable for all one hundred motorcycles and for all one hundred cars. ",

        "date": "2021-12-14T08:24:42.000Z",

        "upVoteNum": 2, //số upvote

        "downVoteNum": 0 //số downvote

    }

}

41. Lấy danh sách câu trả lời

- Mục đích: lấy danh sách câu trả lời (tất cả, theo người dùng và theo post)

- Phương thức: GET

- URL:

+ Lấy tất cả:

localhost:3001/answer/list?page={page}&perPage={perPage}&orderBy={orderColumn}&orderType={orderType}

ví dụ localhost:3001/answer/list?page=1&perPage=2&orderBy=upVoteNum&orderType=desc

+ Lấy theo postId:

localhost:3001/answer/list?page={page}&perPage={perPage}&postId={postId}&orderBy={orderColumn}&orderType={orderType}

ví dụ: localhost:3001/answer/list?page=1&perPage=2&postId=1&orderBy=date&orderType=desc

+ Lấy theo userId:

localhost:3001/answer/list?page={page}&perPage={perPage}&userId={userId}&orderBy={orderColumn}&orderType={orderType}

ví dụ: localhost:3001/answer/list?page=1&perPage=2&userId=1&orderBy=upVoteNum&orderType=desc

- Dữ liệu gửi lên trong params (đã trong URL rồi)

+ userId: id của người dùng cần lấy list answer

+ postId: id của post cần lấy list answer

+ page: page cần lấy

+ perPage: số lượng mỗi tag trong 1 page

+ orderBy: sắp xếp theo cột nào (mặc định nếu không truyền gì là Id của post), các cột dùng được là

. date: thời gian tạo answer

. upVoteNum: số upVote (dùng tạm cái này để sort theo vote, khi nào tôi làm được đoạn trừ hai cái up downvote đi để tính vote thì tôi sẽ nói)

. downVoteNum: số downVote

. …

+ orderType: loại sắp xếp, chọn ‘asc’ hoặc ‘desc’ (mặc định là asc)

- Dữ liệu trả về: cho ví dụ 2 lấy theo postId, mấy cái kia tương tự thế

{

    "success": **true**,

    "result": {

        "data": [

            {

                "Id": 11,

                "userId": 1,

                "postId": 1,

                "answerDetail": "test tạo answer mới nè",

                "date": "2021-12-14T09:06:54.000Z",

                "upVoteNum": 0,

                "downVoteNum": 0

            },

            {

                "Id": 1,

                "userId": 1,

                "postId": 1,

                "answerDetail": " And there is only one entry in the NodeMachine table - the K1 wheel, which is suitable for all one hundred motorcycles and for all one hundred cars. ",

                "date": "2021-12-14T08:24:42.000Z",

                "upVoteNum": 1,

                "downVoteNum": 0

            }

        ],

        "pagination": {

            "total": 3,

            "lastPage": 2,

            "perPage": 2,

            "currentPage": 1,

            "from": 0,

            "to": 2

        }

    }

}

42. Lấy số lượng upvote và downvote cho câu trả lời

- Mục đích: Lấy số lượng upvote và downvote theo answerId

- Phương thức: GET

- URL: localhost:3001/answer/vote-num?answerId={answerId}, ví dụ localhost:3001/answer/vote-num?answerId=1

- Dữ liệu trả về

{

    "success": **true**,

    "result": {

        "upVote": 1, //số lượng upvote

        "downVote": 0 //số lượng downvote

    }

}

43. Lấy vote của user theo answerId

- Mục đích: xem user hiện tại đã vote cho answer id này chưa

- Phương thức: POST

- URL: localhost:3001/answer/user/get-vote

- Dữ liệu gửi lên trong body

{

    "token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJhaG5oZWV5ZW9uMjEwOEBnbWFpbC5jb20iLCJnb29nbGVJZCI6IjEwOTM1MDQ0NTg1MzUzMzgzMDAwMCIsImlhdCI6MTYzOTA1NDE2NiwiZXhwIjoxNjM5MDcyMTY2fQ.zaOx5DfS5XEcUDfkLU-KeAtDKMYCepmyqycIfaRzMfY",

    "answerId": 1 //id của câu trả lời

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": [ //ko tìm thấy thì trả về rỗng

        {

            "answerId": 1,

            "userId": 1,

            "voteType": 1,

            "date": "2021-12-09T07:17:10.000Z"

        }

    ]

}

44. Tạo vote cho câu trả lời

- Mục đích: cho phép người dùng vote cho câu trả lời theo answer Id

- Phương thức: POST

- URL: localhost:3001/answer/user/create-vote

- Dữ liệu gửi lên trong body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJhaG5oZWV5ZW9uMjEwOEBnbWFpbC5jb20iLCJnb29nbGVJZCI6IjEwOTM1MDQ0NTg1MzUzMzgzMDAwMCIsImlhdCI6MTYzOTA1NDE2NiwiZXhwIjoxNjM5MDcyMTY2fQ.zaOx5DfS5XEcUDfkLU-KeAtDKMYCepmyqycIfaRzMfY",

    "answerId": 8,

    "voteType": **true //true là upvote, false là downvote**

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": [

        {

            "answerId": 8,

            "userId": 1,

            "voteType": 1,

            "date": "2021-12-09T13:02:46.000Z"

        }

    ]

}

45. Xoá vote cho câu trả lời

- Mục đích: cho phép người dùng hiện tại xoá vote cho câu trả lời

- Phương thức: POST

- URL: localhost:3001/answer/user/delete-vote

- Dữ liệu gửi lên trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJhaG5oZWV5ZW9uMjEwOEBnbWFpbC5jb20iLCJnb29nbGVJZCI6IjEwOTM1MDQ0NTg1MzUzMzgzMDAwMCIsImlhdCI6MTYzOTA1NDE2NiwiZXhwIjoxNjM5MDcyMTY2fQ.zaOx5DfS5XEcUDfkLU-KeAtDKMYCepmyqycIfaRzMfY",

    "answerId": 8

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": 1

}

45. Tạo câu trả lời

- Mục đích: tạo câu trả lời mới cho 1 bài đăng nào đó

- Phương thức: POST

- URL: localhost:3001/answer/create-answer

- Dữ liệu gửi lên trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTQ4OTMyMiwiZXhwIjoxNjM5NTA3MzIyfQ.vow3b0suHlY7iqnIvuivkNAmn-xM0wIU0SnSgh58b1M",

    "postId": 1,

    "answerDetail": "test tạo answer mới nè"

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 12,

        "userId": 1,

        "postId": 1,

        "answerDetail": "test tạo answer mới nè",

        "date": "2021-12-14T13:46:16.000Z",

        "upVoteNum": 0, //số lượng upvote

        "downVoteNum": 0 //số lượng downvote

    }

}

46. Sửa câu trả lời

- Mục đích: sửa câu trả lời theo answerId

- Phương thức: POST

- URL: localhost:3001/answer/edit-answer

- Dữ liệu gửi lên trong body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTQ4OTMyMiwiZXhwIjoxNjM5NTA3MzIyfQ.vow3b0suHlY7iqnIvuivkNAmn-xM0wIU0SnSgh58b1M",

    "answerId": 12,

    "answerDetail": "Test sửa có được không"

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": {

        "Id": 12,

        "userId": 1,

        "postId": 1,

        "answerDetail": "Test sửa có được không",

        "date": "2021-12-14T13:46:16.000Z",

        "upVoteNum": 0,

        "downVoteNum": 0

    }

}

47. Xoá câu trả lời

- Mục đích: xoá câu trả lời theo answerId

- Phương thức: POST

- URL: localhost:3001/answer/delete-answer

- Dữ liệu gửi lên trong body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJhaG5oZWV5ZW9uMjEwOEBnbWFpbC5jb20iLCJnb29nbGVJZCI6IjEwOTM1MDQ0NTg1MzUzMzgzMDAwMCIsImlhdCI6MTYzOTA1NDE2NiwiZXhwIjoxNjM5MDcyMTY2fQ.zaOx5DfS5XEcUDfkLU-KeAtDKMYCepmyqycIfaRzMfY",

    "answerId": 14

}

- Dữ liệu trả về:

{

    "success": **true**,

    "result": 1 //số bản ghi tác động lên

}

48. Set câu trả lời đúng cho post

- Mục đích: cho phép set câu trả lời đúng cho post hiện tại

- Phương thức: POST

- URL: localhost:3001/post/set-right-answer

- Dữ liệu gửi lên trong body

{

    "token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTQ4OTMyMiwiZXhwIjoxNjM5NTA3MzIyfQ.vow3b0suHlY7iqnIvuivkNAmn-xM0wIU0SnSgh58b1M",

    "postId": 1, //postId cần set câu trả lời đúng

    "answerId": 12 //id của cuâ trả lời (phải đúng của post đấy, không thể set câu trả lời của post khác làm câu trả lời đúng cho post này được

}

- Dữ liệu trả về

{

    "success": **true**,

    "result": {

        "Id": 1,

        "userId": 1,

        "postName": "How interlink external hyperlink/hotspot from WPF with internal hotspot on 3D object in Unity?",

        "postDetail": "Google polyline decoded is not what expected",

        "date": "2021-12-14T08:24:42.000Z",

        "rightAnswerID": 12,

        "viewNum": 1,

        "postUserName": "admin01",

        "numAnswer": 0,

        "upVoteNum": 0,

        "downVoteNum": 0,

        "likeNum": 0

    }

}

49. Xoá câu trả lời đúng

- Mục đích: xoá câu trả lời đúng khỏi post theo postId

- Phương thức: POST

- URL: localhost:3001/post/delete-right-answer

- Dữ liệu gửi lên trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTAzMjY3MiwiZXhwIjoxNjM5MDUwNjcyfQ.R\_hnZcIVvAbceX8DR3tSy-XF4yqCGSbbkb\_UN0R6IKI",

    "postId": 1 //post id cần xoá

}

- Dữ liệu trả về: trả về thông tin cơ bản của post sau khi sửa

{

    "success": **true**,

    "result": {

        "Id": 1,

        "userId": 1,

        "postName": "How interlink external hyperlink/hotspot from WPF with internal hotspot on 3D object in Unity?",

        "postDetail": "Google polyline decoded is not what expected",

        "date": "2021-12-14T08:24:42.000Z",

        "rightAnswerID": **null**,

        "viewNum": 1,

        "postUserName": "admin01",

        "numAnswer": 0,

        "upVoteNum": 0,

        "downVoteNum": 0,

        "likeNum": 0

    }

}

50 Lấy danh sách favorite post của người dùng

- Mục đích: lấy danh sách về favorite post của người dùng có sắp xếp và phân trang

- Phương thức: GET

- URL

localhost:3001/post/user/get-favorite-posts?page={page}&perPage={perPage}&orderBy={column}&orderType={orderType}

Ví dụ

localhost:3001/post/user/get-favorite-posts?page=1&perPage=2&orderBy=addToFavoritePostDate&orderType=desc

- Dữ liệu gửi lên

+ Trong body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTg5ODI5OCwiZXhwIjoxNjM5OTE2Mjk4fQ.gWmMErKiqVPNxK9k3sMOI-fp77AnIS1PIUSgxtBt07A" //access token của ngưởi dùng

}

+ Trong param

+ page: page cần lấy

+ perPage: số lượng mỗi tag trong 1 page

+ orderBy: sắp xếp theo cột nào (mặc định nếu không truyền gì là Id của post), các cột dùng được là

. date: thời gian tạo post

. upVoteNum: số upVote (dùng tạm cái này để sort theo vote, khi nào tôi làm được đoạn trừ hai cái up downvote đi để tính vote thì tôi sẽ nói)

. downVoteNum: số downVote

. addToFavoritePostDate: thời gian mà user thêm post này vào sanh sách post yêu thích

. …(tất cả các trường trong mỗi post trả về đều dùng được)

+ orderType: loại sắp xếp, chọn ‘asc’ hoặc ‘desc’ (mặc định là asc)

51. Get like of post

- Mục đích: lấy like của người dùng nếu họ đã like post đó, ko có thỉ trả về mảng rỗng

- Phương thức: GET

- URL: localhost:3001/post/user/get-like

- Dữ liệu gửi lên trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTg5ODI5OCwiZXhwIjoxNjM5OTE2Mjk4fQ.gWmMErKiqVPNxK9k3sMOI-fp77AnIS1PIUSgxtBt07A",

    "postId": 4 //id của post cần lấy

}

- Dữ liệu trả về

{

    "success": **true**,

    "result": [ //khi không có like thì trả về mảng rỗng

        {

            "postId": 4,

            "userId": 1,

            "date": "2021-12-18T06:46:48.000Z"//ngày post vào favorite post

        }

    ]

}

52. Add post to favorite post

- Mục đích: cho user thêm post vào sanh sách yêu thích

- Phương thức: POST

- URL: localhost:3001/post/user/add-favorite-post

- Dữ liệu gửi lên trong body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTg5ODI5OCwiZXhwIjoxNjM5OTE2Mjk4fQ.gWmMErKiqVPNxK9k3sMOI-fp77AnIS1PIUSgxtBt07A",

    "postId": 2 //id của post cần thêm

}

- Dữ liệu trả về

{

    "success": **true**,

    "result": [

        {

            "postId": 2,

            "userId": 1,

            "date": "2021-12-19T07:31:04.000Z" // ngày post vào favorite post

        }

    ]

}

53. Xoá post khỏi favorite post

- Mục đích: xoá post khỏi danh sách post yêu thích

- Phương thức: POST

- URL: localhost:3001/post/user/delete-favorite-post

- Dữ liệu gửi lên trong body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJJZCI6MSwicm9sZSI6MSwiZW1haWwiOiJ2dXF1YW5naHV5MjEwODE5OTlAZ21haWwuY29tIiwiZ29vZ2xlSWQiOm51bGwsImlhdCI6MTYzOTg5ODI5OCwiZXhwIjoxNjM5OTE2Mjk4fQ.gWmMErKiqVPNxK9k3sMOI-fp77AnIS1PIUSgxtBt07A",

    "postId": 2

}

- Dữ liệu trả về

{

    "success": **true**,

    "result": 1 //số bản ghi bị tác động

}