7. 會員系統(API製作)

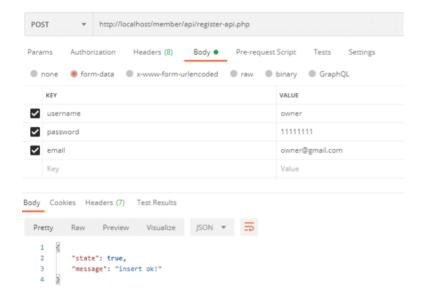
資料庫結構



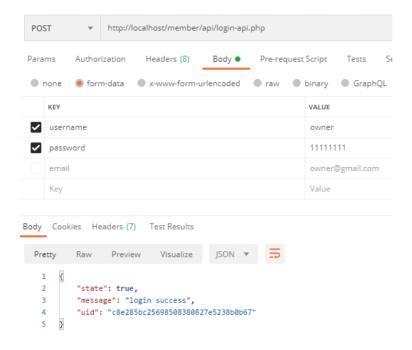
編號	API	INPUT	ОИТРИТ	說明	回復訊息
1	register-api	username password email	"state": true or false "message": "insert ok! " or error message	新增註冊帳號	{ "state": true, "message": "insert ok!" }
2	register-check-uni- api	username	"state": true or false "message": "username already exist"	註冊時確認帳號是否已存在(表示登入中)	{ "state": true, "message": "帳號已存在,不可使用!" } { "state": false, "message": "帳號不存在,可以使用!" }
3	login-api	username password	"state": true or false "message": "login success" "uid":	登入帳號若成功產生 一組新的uid	{ "state": true, "message": "login success", "uid": "340cdf004e0e2e12c473679d7eede774" } { "state": false, "message": "login error: "SELECT * FROM userdata WHERE Username = 'owners999' && Password = '1bbd886460827035e5d605ed44252251' }
4	check-user-api	cookie_ui d	"state": true or false "message": "username already login or not", "data":使用者的所有資訊	判斷uid是否已存在	<pre>{ "state": true, "message": "username already login", "data": { "ID": "35", "Username": "owner99", "Email": "owner009@gmail.com", "Uid": "340cdf004e0e2e12c473679c7eede774", "State": "1" } }</pre>

					"state": false, "message": "username does not login" }
5	update-api	id email	"state": true or false "message": "update success or failure"	更新email	{ "state": true, "message": "update success" } { "state": false, "message": "update failure" }
6	del-api	id	"state": true or false "message": "delete success or failure"	刪除會員	{ "state": true, "message": "delete success" } { "state": false, "message": "delete failure" }
7	logout-api	id	"state": true or false message": "logout success or logout failure"	登出會員	{ "state": true, "message": "logout success" } { "state": false, "message": "logout failure"UPDATE us erdata SET State = false, Uid = " WHERE ID = '31' }
8	member-list-api	none	"state": true, "data":' 使用者列表	會員列表	{ "state": true, "data": [{ "ID": "33", "Username": "owner09", "Email": "owner009@gmail.com", "Uid": "", "State": "0" }, { "ID": "32", "Username": "owner", "Email": "owner009@gmail.com", "Uid": "", "State": "0" }] }

1. register-api



2. login-api



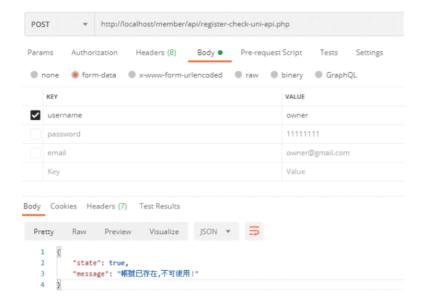
登入成功後會產生新的Uid並將state設定為1(登入狀態)



3. register-check-uni-api

註冊時判斷帳號是否已存在!

Vue 第4頁



4. check-user-api

判斷uid是否已存在若存在表示登入中,並回傳該使用者資料!

```
(?php
if(isset($_POST['cookie_uid'])){
    //cookie_uid 用来比較常品是分
    $p_cookie_uid = $_POST['cookie_uid'];
    require_once("dotools.inc.php");

$link = create connection();

$sql = "SetLet To, Username, Email, Uid, State FROM userdata NHERE Uid = '$p_cookie_uid'";

$result = execute_sql($link, "member", $sql);

//用来輸出json 以便能示金入資訊
$myarray = array();

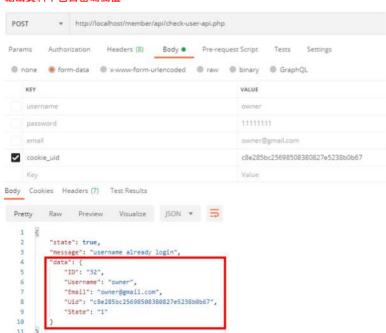
$myarray = mysqli_fetch_assoc($result);

if(mysqli_num_rows($result) == 1){
    echo '{
        "state": true,
        "message": "username already login",
        "data" : '.json_encode($myarray).'
    }';

}else{
    echo '{
        "state": false,
        "message": "username does not login"
    }';

}else{
    echo '{
        "state": false,
        "message": "column cannt null!"
    }';
}
```

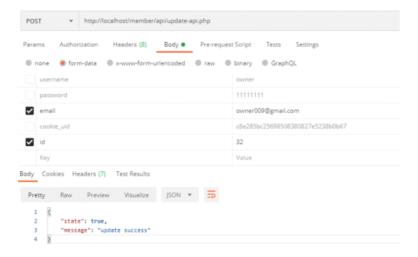
輸出資料不包含密碼欄位



Vue 第 5 頁

5. update-api

更新除了判斷語法是否正確以外還需要利用mysqli affected rows()判斷是否有執行紀錄!

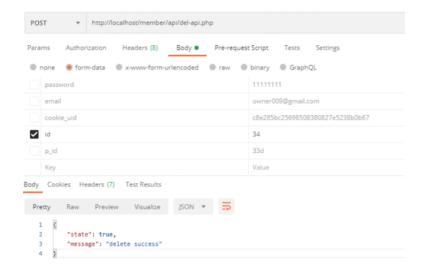


6. del-api

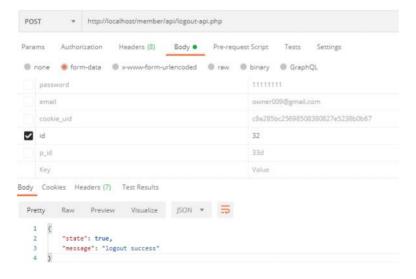
刪除除了判斷語法是否正確以外還需要利用mysqli_affected_rows()判斷是否有執行紀錄!

```
if(isset($_POST['id'])){
    $p_id = $_POST['id'];
    require_once("dbtools.inc.php");

    $link = create_connection();
    $sql = "DELETE FROM userdata WHERE id = '$p_id'";
    //delete需要再判断mysqli_affected_rows数目,才可確認是否删除成功
    if(execute_sql($link, "member", $sql) && mysqli_affected_rows($link) == 1){
        echo '{
            "state": true,
            "message": "delete success"
        }';
        mysqli_close($link);
    }else{
        echo '{
            "state": false,
            "message": "delete failure"
        }';
    }
}else{
    echo '{
        "state": false,
        "message": "id cannt be null!"
        }';
}
```



7. logout-api



登出後清空Uid並將State改為false

```
        ←丁十
        ▼ ID
        Username
        Password
        Email
        Created_at
        Uid
        State

        □ ● 報報
        ● 複数
        ● 複数</
```

8. member-list-api

Vue 第7頁

```
hp
require_once("dbtools.inc.php");
$link = create_connection();
$sql = "SELECT ID, Username, Email, Uid, State FROM userdata ORDER BY ID DESC";
$result = execute_sql($link, "member", $sql);
if(mysqli_num_rows($result) >0){
    $myarray = array();
    while($row = mysqli_fetch_assoc($result)){
        $myarray[] = $row;
}
             getho '{
    "state": true,
    "data": '.json_encode($myarray).'
    "to"
}';
}else(
echo '{
    "state": false,
    "message":"no data"
  GET • http://localhost/member/api/member-list-api.php
 Params Authorization Headers (6) Body Pre-request Script Tests Settings
 none form-data x-www-form-urlencoded raw binary GraphQL
                                                                                             This request does n
Body Cookies Headers (7) Test Results
 Pretty Raw Preview Visualize JSON ▼ 👼
          {
    1
                 "state": true,
     2
                 "data": [
     3
                  data":
{
    "ID": "33",
    "Username": "owner09",
    "Email": "owner009@gmail.com",
    "Uid": "",
    "State": "0"
     5
     6
     8
                   },
   10
   11
                         "ID": "32",
"Username": "owner",
"Email": "owner@09@gmail.com",
"Uid": "",
"State": "@"
   12
    13
    15
    17
    18
                1
          )
```