

7. 會員系統(API製作)

資料庫結構

#	名稱	類型	編碼與排序	屬性	空值(Null)	預設值	備註	額外資訊	動作
<input type="checkbox"/> 1	ID	int(11)			否	無		AUTO_INCREMENT	修改 刪除 更多
<input type="checkbox"/> 2	Username	varchar(32) utf8_unicode_ci			否	無			修改 刪除 更多
<input type="checkbox"/> 3	Password	varchar(32) utf8_unicode_ci			否	無			修改 刪除 更多
<input type="checkbox"/> 4	Email	varchar(50) utf8_unicode_ci			否	無			修改 刪除 更多
<input type="checkbox"/> 5	Created_at	timestamp			否	CURRENT_TIMESTAMP			修改 刪除 更多
<input type="checkbox"/> 6	Uid	varchar(32) utf8_unicode_ci			否	無	cookie判斷是否登入		修改 刪除 更多
<input type="checkbox"/> 7	State	tinyint(1)			否	0	登入狀態		修改 刪除 更多

編號	API	INPUT	OUTPUT	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 = 'owner999' && Password = '1bbd886460827015e5d605ed44252251'" }
4	check-user-api	cookie_uid	"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	{ "state": true, "message": "username already login", "data": { "ID": "35", "Username": "owner99", "Email": "owner009@gmail.com", "Uid": "340cdf004e0e2e12c473679d7eede774", "State": "1" } } {

					<pre> "state": false, "message": "username does not login" } </pre>
5	update-api	id email	<pre> "state": true or false "message": "update success or failure" </pre>	更新email	<pre> { "state": true, "message": "update success" } { "state": false, "message": "update failure" } </pre>
6	del-api	id	<pre> "state": true or false "message": "delete success or failure" </pre>	刪除會員	<pre> { "state": true, "message": "delete success" } { "state": false, "message": "delete failure" } </pre>
7	logout-api	id	<pre> "state": true or false message": "logout success or logout failure" </pre>	登出會員	<pre> { "state": true, "message": "logout success" } { "state": false, "message": "logout failure"UPDATE us erdata SET State = false, Uid = " WHERE ID = '31' } </pre>
8	member-list-api	none	<pre> "state": true, "data": 使用者列表 </pre>	會員列表	<pre> { "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" }] } </pre>

1. register-api

```

<?php
//註冊資料不許為空
if(isset($_POST['username']) && isset($_POST['password']) && isset($_POST['email'])){
    $p_username = $_POST['username'];
    $p_password = $_POST['password'];
    $p_email = $_POST['email'];
}

```


POST http://localhost/member/api/login-api.php

Params Authorization Headers (8) **Body** Pre-request Script Tests St

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

KEY	VALUE
<input checked="" type="checkbox"/> username	owner
<input checked="" type="checkbox"/> password	11111111
<input type="checkbox"/> email	owner@gmail.com
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "state": true,
3   "message": "login success",
4   "uid": "c8e285bc25698508380827e5238b0b67"
5 }
```

登入成功後會產生新的Uid並將state設定為1(登入狀態)

資料庫: 127.0.0.1 » 資料庫: member » 資料庫: userdata

顯示 0 - 0 列 (總計 1 筆, 查詢用了 0.0008 秒。)

SELECT * FROM 'userdata'

全部顯示 | 資料列數: 25 | 篩選資料列: 搜尋此資料表

ID	Username	Password	Email	Created_at	Uid	State
32	owner	1bbd886460827015e5d605ed44252251	owner@gmail.com	2020-12-22 13:28:15	c8e285bc25698508380827e5238b0b67	1

全部顯示 | 資料列數: 25 | 篩選資料列: 搜尋此資料表

3. register-check-uni-api

註冊時判斷帳號是否已存在!

```
<?php
if(isset($_POST['username'])){
    $p_username = $_POST['username'];

    require_once("dbtools.inc.php");

    $link = create_connection();

    $sql = "SELECT * FROM userdata WHERE Username = '$p_username'";
    $result = execute_sql($link, "member", $sql);

    if(mysqli_num_rows($result) == 1){
        echo '{
            "state": true,
            "message": "帳號已存在,不可使用!"
        }';
        mysqli_close($link);
    }else{
        echo '{
            "state": false,
            "message": "帳號不存在,可以使用!"
        }';
    }
}else{
    echo '{
        "state": false,
        "message": "column cannt null!"
    }';
}
}
```

POST http://localhost/member/api/register-check-uni-api.php

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

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

KEY	VALUE
<input checked="" type="checkbox"/> username	owner
<input type="checkbox"/> password	11111111
<input type="checkbox"/> email	owner@gmail.com
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "state": true,
3   "message": "帳號已存在,不可使用!"
4 }

```

4. check-user-api

判斷uid是否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 用來比對是否已登入
    $p_cookie_uid = $_POST['cookie_uid'];
    require_once("dbtools.inc.php");

    $link = create_connection();
    $sql = "SELECT ID, Username, Email, Uid, State FROM userdata WHERE Uid = '$p_cookie_uid'";
    $result = execute_sql($link, "member", $sql);

    //用來輸出json 以便顯示登入資訊
    $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

POST http://localhost/member/api/check-user-api.php

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

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

KEY	VALUE
<input type="checkbox"/> username	owner
<input type="checkbox"/> password	11111111
<input type="checkbox"/> email	owner@gmail.com
<input checked="" type="checkbox"/> cookie_uid	c8e285bc25698508380827e5238b0b67
Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "state": true,
3   "message": "username already login",
4   "data": {
5     "ID": "32",
6     "Username": "owner",
7     "Email": "owner@gmail.com",
8     "Uid": "c8e285bc25698508380827e5238b0b67",
9     "State": "1"
10  }
11 }

```

5. update-api

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

```
<?php
if(isset($_POST['id'])){
    $p_id = $_POST['id'];
    $p_email = $_POST['email'];

    require_once("dbtools.inc.php");

    $link = create_connection();

    $sql = "UPDATE userdata SET Email='$p_email' WHERE ID = '$p_id'";
    //update需要再判斷mysql_affected_rows數目, 才可確認是否更新成功
    if(execute_sql($link, "member", $sql) && mysql_affected_rows($link) == 1){
        echo '{
            "state": true,
            "message": "update success"
        }';
        mysqli_close($link);
    }else{
        echo '{
            "state": false,
            "message": "update failure"
        }';
    }
}else{
    echo '{
        "state": false,
        "message": "id cannt be null!"
    }';
}
?>
```

POST http://localhost/member/api/update-api.php

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

<input type="checkbox"/>	username	owner
<input type="checkbox"/>	password	11111111
<input checked="" type="checkbox"/>	email	owner009@gmail.com
<input type="checkbox"/>	cookie_uid	c8e285bc25698508380627e5238b0b67
<input checked="" type="checkbox"/>	id	32
	Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "state": true,
3   "message": "update success"
4 }
```

6. del-api

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

```
<?php
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需要再判斷mysql_affected_rows數目, 才可確認是否刪除成功
    if(execute_sql($link, "member", $sql) && mysql_affected_rows($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!"
    }';
}
?>
```

POST http://localhost/member/api/del-api.php

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

password 11111111

email owner009@gmail.com

cookie_uid c8e285bc25698508380827e5238b0b67

☒ id 34

☐ p_id 33d

Key Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "state": true,
3   "message": "delete success"
4 }

```

7. logout-api

```

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

    $link = create_connection();
    //清空 Uid 和 State 表示登出
    $sql = "UPDATE userdata SET State = false, Uid = '' WHERE ID = '$p_id'";
    //update需要再判斷mysqli_affected_rows數目, 才可確認是否更新成功
    if(execute_sql($link, "member", $sql) && mysqli_affected_rows($link) == 1){
        echo '{
            "state": true,
            "message": "logout success"
        }';
        mysqli_close($link);
    }else{
        echo '{
            "state": false,
            "message": "logout failure". $sql.'
        }';
    }
}else{
    echo '{
        "state": false,
        "message": "id cannt be null!"
    }';
}
}
?>

```

POST http://localhost/member/api/logout-api.php

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

password 11111111

email owner009@gmail.com

cookie_uid c8e285bc25698508380827e5238b0b67

☒ id 32

☐ p_id 33d

Key Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "state": true,
3   "message": "logout success"
4 }

```

登出後清空Uid並將State改為false

	ID	Username	Password	Email	Created_at	Uid	State
	32	owner	1bbd086460827015e5d605e44252251	owner009@gmail.com	2020-12-22 13:28:15		0
	33	owner09	1bbd086460827015e5d605e44252251	owner009@gmail.com	2020-12-22 14:04:10		0

8. member-list-api


```

<?php
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[] = $row;
    }
    echo '{
        "state": true,
        "data": ' . json_encode($myarray) . '
    }';
}else{
    echo '{
        "state": false,
        "message": "no data"
    }';
}
mysqli_close($link);
?>

```

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 not have a body

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2      "state": true,
3      "data": [
4          {
5              "ID": "33",
6              "Username": "owner09",
7              "Email": "owner009@gmail.com",
8              "Uid": "",
9              "State": "0"
10         },
11         {
12             "ID": "32",
13             "Username": "owner",
14             "Email": "owner009@gmail.com",
15             "Uid": "",
16             "State": "0"
17         }
18     ]
19 }

```


