

# 分散式系統

## Lab: RESTful Web Services

學號:108753208

系級:資科碩

姓名:葉冠宏

部落中需要更多的支援而調動了許多野豬騎士。  
我們需要存取目前野豬騎士(hogRider)的資訊，基於給定的樣板 restfulServer.js，完成一個野豬騎士的 RESTful Server。



### 1.GET

請貼上 test-GET-tom.js 的程式

答:

```
const client = require('node-fetch');

(async () => {
  const resp = await client('http://localhost:3000/hogRider/tom', {
    method: 'GET'
  });

  const data = await resp.json();
  console.log(data);
})();
```

請貼上 test-GET-tom.js 執行後所印出的 data 內容

答:

```
C:\Users\chrysal212\Desktop\hw2\restlab>node test-GET-tom.js
{ name: 'tom', age: 19, attack: 105, defense: 90 }
```

修改 test-GET-tom.js 程式，尋找一個不存在的人，例如:

http://localhost:3000/hogRider/mary，測試看看是否輸出第 3 步驟的內容  
( { "error": "not found" } )

答:

```
const client = require('node-fetch');

(async () => {
  const resp = await client('http://localhost:3000/hogRider/marry', {
    method: 'GET'
  });

  const data = await resp.json();
  console.log(data);
})();
```

```
C:\Users\chrysal212\Desktop\hw2\restlab>node test-GET-error.js
{ error: 'not found' }
```

## 2.POST

請在下面貼上 test-POST.js 的內容:

答:

```
const client = require('node-fetch');

(async () => {
  const resp = await client('http://localhost:3000/hogRider', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      name : "mary",
      age : 17,
      attack : 99,
      defense : 99
    })
  });

  const data = await resp.json();
  console.log(data);
})();
```

使用 test-GET.js 會向 url `http://localhost:3000/hogRider` 下達 GET，可用來列出所有騎士資料，觀察 mary 是否順利新增。

答:

```
C:\Users\chrysal212\Desktop\hw2\restlab>node test-POST.js
{ count: 3 }
```

```
C:\Users\chrysal212\Desktop\hw2\restlab>node test-GET.js
[
  { name: 'john', age: 18, attack: 100, defense: 100 },
  { name: 'tom', age: 19, attack: 105, defense: 90 },
  { name: 'mary', age: 17, attack: 99, defense: 99 }
]
```

### 3.PUT

寫作 test-PUT.js 來驗證結果，請在下面貼上 test-PUT.js 的內容：

答：

```
const client = require('node-fetch');

(async () => {
  const resp = await client('http://localhost:3000/hogRider/tom', {
    method: 'PUT',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({
      name: "tom",
      age: 99,
      attack: 0,
      defense: 0
    })
  });

  const data = await resp.json();
  console.log(data);
})();
```

```
C:\Users\chrystal212\Desktop\hw2\restlab>node test-PUT.js
{ name: 'tom', age: 99, attack: 0, defense: 0 }
```

### 4.

請重啟 restful-server.js，之後依序執行 test-POST.js->test-PUT.js->test-GET.js 將輸出結果貼在下方：

答：

```
C:\Users\chrystal212\Desktop\hw2\restlab>node test-POST.js
{ count: 3 }

C:\Users\chrystal212\Desktop\hw2\restlab>node test-PUT.js
{ name: 'tom', age: 99, attack: 0, defense: 0 }

C:\Users\chrystal212\Desktop\hw2\restlab>node test-GET.js
[
  { name: 'john', age: 18, attack: 100, defense: 100 },
  { name: 'tom', age: 99, attack: 0, defense: 0 },
  { name: 'mary', age: 17, attack: 99, defense: 99 }
]
```

將您的 restful-server.js 中的所有程式碼貼在下方：

答：

```

restful-server.js
1  const server = require('fastify')();
2
3  let john = {
4    name: "john",
5    age: 18,
6    attack: 100,
7    defense: 100
8  };
9
10 let tom = {
11   name: "tom",
12   age: 19,
13   attack: 105,
14   defense: 90
15 };
16
17 let hogRiders = [john, tom];
18
19 server.get('/hogRider', function (req, res) {
20   return hogRiders;
21 });
22
23 server.get('/hogRider/:name', function (req, res) {
24   // 請依Lab說明寫作
25   let result = hogRiders.find(element => element.name === req.params.name);
26
27   if (result){
28     res.status(200).send(result);
29   }else{
30     res.status(400).send({"error":"not found"});
31   }
32 });
33
34 server.post('/hogRider', function (req, res) {
35   // 請依Lab說明寫作
36   let newRider=req.body;
37   hogRiders.push(newRider);
38   res.status(200).send({"count":hogRiders.length});
39 });
40
41 server.put('/hogRider/:name', function (req, res) {
42   // 請依Lab說明寫作
43   let index = hogRiders.findIndex(element => element.name === req.params.name);
44   hogRiders[index] = req.body;
45   let up=hogRiders[index];
46   res.status(200).send(up);
47 });
48
49 server.listen(3000, "127.0.0.1");

```