

---

# RESTful 操作手冊

---

# Outline

---

2

- **RESTful Service 功能**
  - **RESTful Service 資源操作定義**
  - **RESTful Service 功能操作範例**
-

# Outline

---

3

- **RESTful Service 功能**
  - RESTful Service 定義
  - RESTful Service 功能操作範例
-

# RESTful Service 功能

---

4

- 從Database中的Table**取得**全部資料: SELECT FROM
- 從Database中的Table**取得**特定資料: SELECT FROM WHERE
- 於Database中的Table**新增**資料: INSERT INTO TABLE
- 於Database中**新增**Table: CREATE TABLE
- 於Database中的Table**更新**單筆資料: UPDATE WHERE
- 於Database中的Table**刪除**資料: DELETE FROM、DROP TABLE

※後續將以DB表示Database

---

# Outline

---

5

- RESTful Service 功能
  - **RESTful Service 定義**
  - RESTful Service 功能操作範例
-

# RESTful Service 定義

---

6

## ■ URL 定義

➤ 操作什麼資源(What)

- DB、Table

## ■ HTTP 操作方法定義

➤ 如何操作資源(How)

- GET、POST、PUT：取得、新增、更新資料

## ■ Json 定義

➤ 僅用於傳遞資料

## ■ Server Response 定義

➤ 提示操作是否成功

---

# RESTful Service 定義 (Cont.)

---

7

- URL 定義
  - HTTP 操作方法定義
  - Json 格式定義
  - Server Response 定義
-

# RESTful Service 定義 (Cont.)

---

8

## ■ URL 定義

➤  $\text{http://} \overset{\textcircled{1}}{\text{[Service IP]}} \overset{\textcircled{2}}{\text{/[DB Name]}} \text{/[Table Name]}/{\text{Condition}}$

### $\textcircled{1}$ Service IP

- 管理員指定
- 140.116.86.246:9999

### $\textcircled{2}$ DB Name

- 管理員指定
  - 使用者可使用之Database
-



# RESTful Service 定義 (Cont.)

---

9

## ■ URL 定義

➤  $\text{http://}[\text{Service IP}]/[\text{DB Name}]/[\text{Table Name}]/\{\text{Condition}\}$

### 3 Table Name

- 使用者自訂
- 使用者欲存放資料之Table

### 4 Condition

- [Optional]
  - 用於：(1)取得資料；(2)更新資料
  - 格式：[資料庫Table欄名稱]=[欄位值]
  - 若無Condition: 主要針對該表格的進行操作
  - 若有Condition: 主要針對該表格內的資料進行操作
-

# RESTful Service 定義 (Cont.)

---

10

- URL 定義
  - **HTTP 操作方法 定義**
  - Json 格式 定義
  - Server Response 定義
-

# RESTful Service 定義 (Cont.)

---

11

## ■ HTTP操作方法定義

### ➤ GET: 取得全部資料

- GET `http://140.116.86.246:9999/adas/car`
- 從該IP的adas資料庫的car資料表，取得所有資料

### ➤ GET: 取得特定資料

- GET `http://140.116.86.246:9999/adas/car/speed=30`
- 從該IP的adas資料庫的car資料表  
取得欄位"speed"的值為30的資料

Condition



# RESTful Service 定義 (Cont.)

---

12

## ■ HTTP操作方法定義

### ➤ POST: 新增資料、新增Table

- POST `http://140.116.86.246:9999/adas/car` with Json
- Json: 預新增的資料
- 對該IP的adas資料庫的car資料表，插入Json資料

### ➤ PUT: 更新資料

- PUT `http://140.116.86.246:9999/adas/car/id=0x00` with Json
- Json: 預更新的資料
- 對該IP的adas資料庫的car資料表的id=0x00的資料  
根據Json檔案進行更新

Condition



# RESTful Service 定義 (Cont.)

---

13

- URL 定義
  - HTTP 操作方法 定義
  - **Json 格式 定義**
  - Server Response 定義
-

# RESTful Service 定義 (Cont.)

14

## ■ Json格式定義 – 取得、新增資料

```
{  
    一筆資料  
    "[Row Index]": {  
        "[Key]": "[Value]",  
        "[Key]": "[Value]",  
        "[Key]": "[Value]",  
        ...  
    },  
    ...  
}
```

# RESTful Service 定義 (Cont.)

15

## ■ Json格式建議範例 - 取得、新增資料

{

**第一筆資料**  
"0": {  
    "ID": "0x00",  
    "Speed": "70"  
},

**第二筆資料**  
"1": {  
    "ID": "0x11",  
    "Speed": "30"  
}

}

"[Row Index]"

"[Key]": "[Value]"

讀取資料庫

ID	Speed
0x00	70
0x11	30

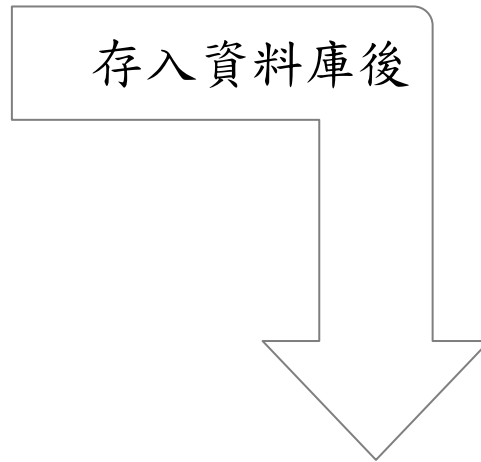
# RESTful Service 定義 (Cont.)

16

## ■ Json格式 不建議 範例 - 取得、新增資料

```
{  
  "0":{  第一筆資料  
    "ID": "0x00",  
    "Speed": {  
      "First": "70",  
      "Second": "50"  
    }  
  },  
  "1":{  第二筆資料  
    "ID": "0x11",  
    "Speed": "30"  
  }  
}
```

存入資料庫後



ID	Speed
0x00	{'second': '50', 'first': '70'}
0x11	30



# RESTful Service 定義 (Cont.)

---

17

## ■ Json格式定義 - 更新資料

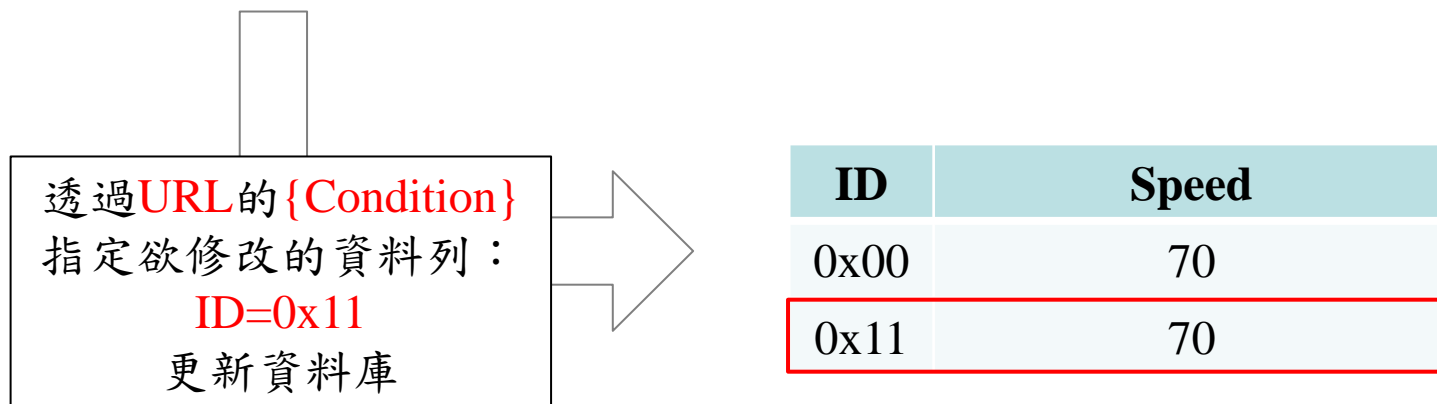
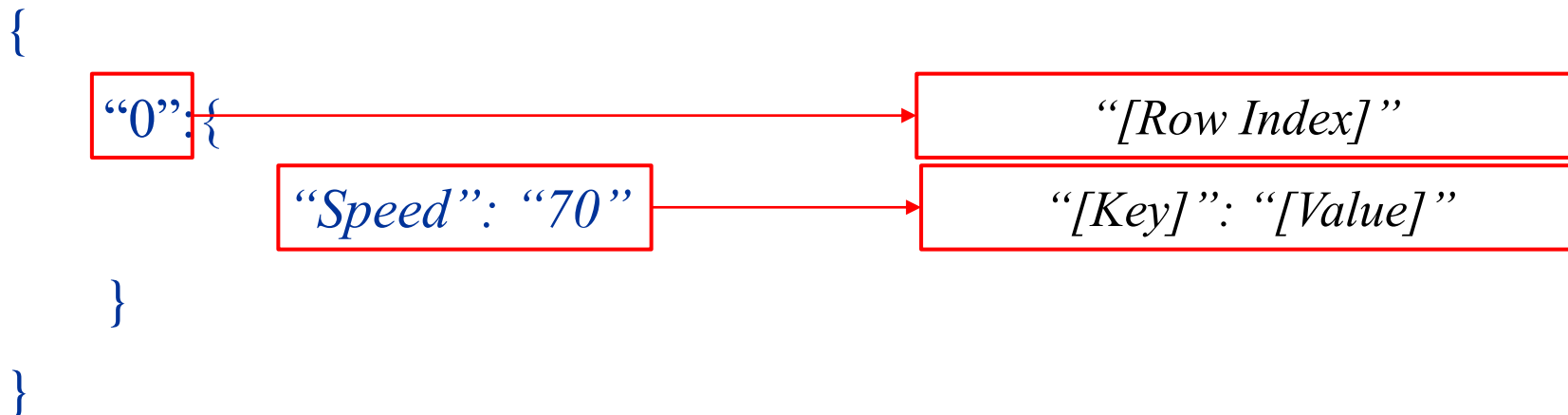
```
{  
    "[Row Index]": {  
        "[Key]": "[Value]",  
        "[Key]": "[Value]",  
        ...  
    }  
}
```

---

# RESTful Service 定義 (Cont.)

18

## ■ Json格式範例 - 更新資料



# RESTful Service 定義 (Cont.)

---

19

- URL 定義
  - HTTP 操作方法 定義
  - Json 格式 定義
  - **Server Response 定義**
-

# RESTful Service 定義 (Cont.)

---

20

## ■ Server Response 定義

### ➤ HTTP GET (取得資料)

- 若操作成功：以Json格式回傳資料
- 若操作失敗：回傳"<Response 500> [錯誤訊息]"

### ➤ HTTP POST、PUT (新增、修改資料)

- 若操作成功：回傳"<Response 200> [成功訊息]"
  - 若操作失敗：回傳"<Response 500> [錯誤訊息]"
-

# RESTful Service 定義 (Cont.)

---

21

## ■ Server Response 定義

### ➤ 操作失敗可能原因

- 使用者輸入之參數錯誤，Ex: URL 錯誤
- 使用者操作之方法錯誤，Ex: HTTP HEAD
- RESTful Service 故障或不支援的功能

# Outline

---

22

- RESTful Service 功能
  - RESTful Service 定義
  - **RESTful Service 功能操作範例**
-

# RESTful Service 功能操作

---

23

## ■ 取得Table全部資料

- User必須指定(1)DB Name；(2)Table Name
  - 使用HTTP GET
    - URL: `http://140.116.86.246:9999/[DB Name]/[Table Name]`
  - 不須傳遞Json檔案
  - Server Response
    - 若操作成功：以Json格式回傳資料
    - 若操作失敗：回傳”<Response 500> [錯誤訊息]”
-

# RESTful Service 功能操作 (Cont.)

25

## ■ 取得Table全部資料範例一

- DB Name: **adas** ; Table Name: **car**
- GET <http://140.116.86.246:9999/adas/car>
- Python 程式碼
  - `r = requests.get("http://140.116.86.246:9999/adas/car")`
  - `r.json()` # 取得RESTful Service回傳之Json之資料
  - **`r.text` # 取得Server Response**

```
>>> r = requests.get("http://140.116.86.246:9999/adas/car")  
>>> r.json()
```

```
{ '1': { 'speed': '30', 'id': '0x11' }, '0': { 'speed': '70', 'id': '0x00' } }
```

Data



# RESTful Service 功能操作 (Cont.)

26

## ■ 取得Table全部資料範例二

- DB Name: **adas** ; Table Name: **car**
- GET <http://140.116.86.246:9999/adas/car>
- 使用瀏覽器
  - 瀏覽器瀏覽網頁的動作為GET



# RESTful Service 功能操作 (Cont.)

---

27

## ■ 取得Table特定資料

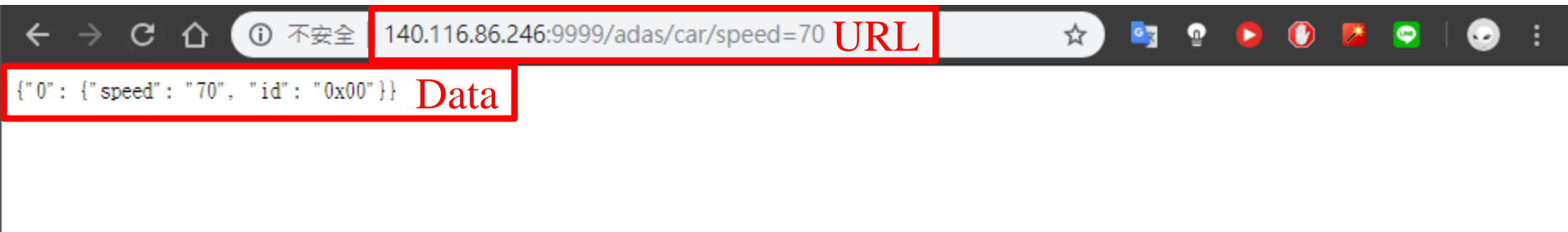
- User必須指定(1)DB Name；(2)Table Name；(3)Condition
  - 使用HTTP GET
    - URL: `http://140.116.86.246:9999/[DB Name]/[Table Name]/[Key]=[Value]`
  - 不須傳遞Json檔案
  - Server Response
    - 若操作成功：以Json格式回傳資料
    - 若操作失敗：回傳”<Response 500> [錯誤訊息]”
-

# RESTful Service 功能操作 (Cont.)

28

## ■ 取得Table特定資料範例

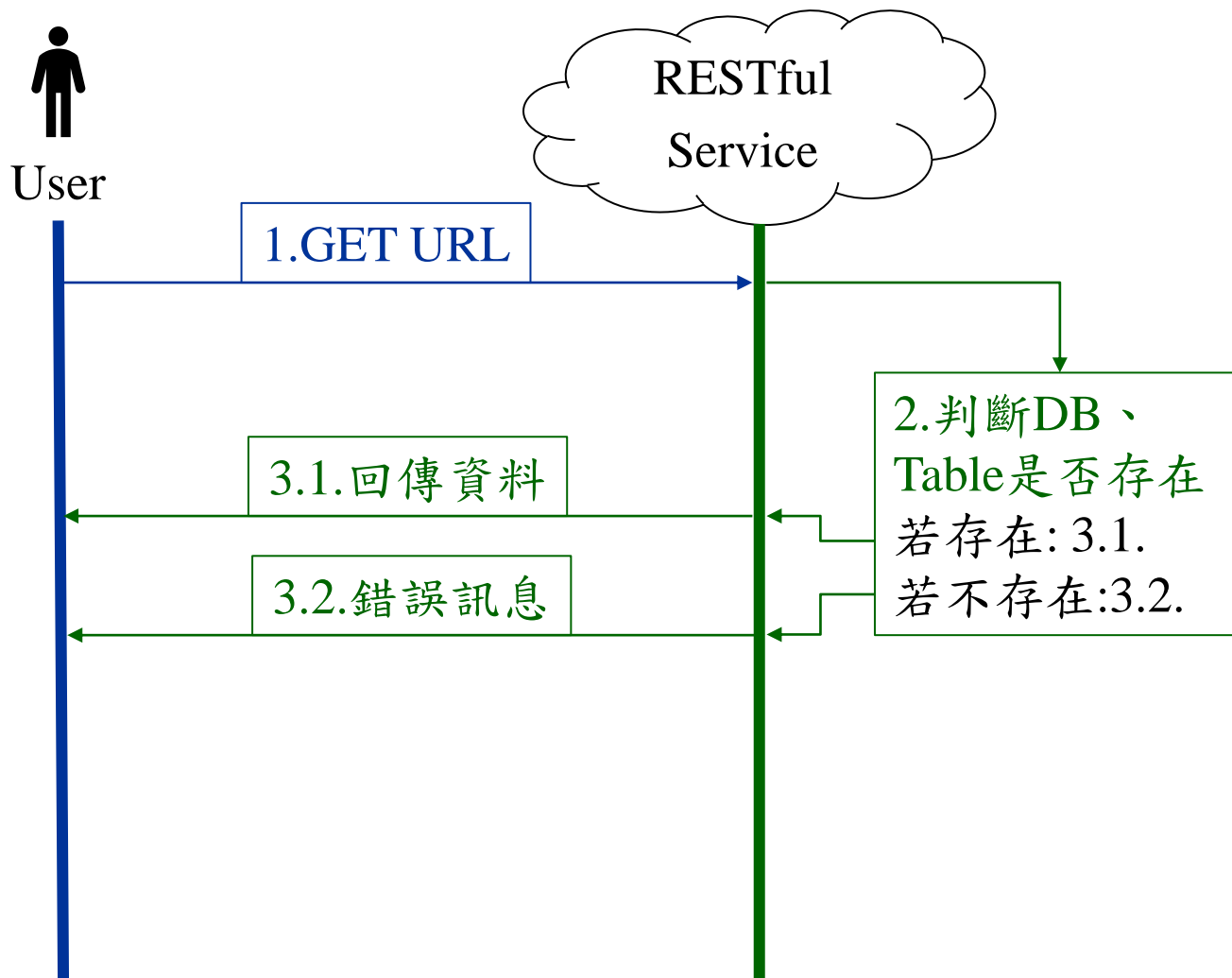
- DB Name: **adas** ; Table Name: **car** ; Condition: **speed=70**
- GET <http://140.116.86.246:9999/adas/car/speed=70>
- 使用瀏覽器瀏覽



# RESTful Service 功能操作

29

## ■ Summery – 取得Table資料



# RESTful Service 功能操作 (Cont.)

---

30

## ■ 新增資料至Table

- ※ RESTful Service不處理「Table資料唯一性」的問題
    - 使用者須自行處理
    - 建議方法：每筆資料加入Primary Key(即Json格式所規定)
  - User必須指定(1)DB Name；(2)Table Name
  - 使用HTTP POST
    - URL: `http://140.116.86.246:9999/[DB Name]/[Table Name]`
  - 將欲新增之資料格式化為Json檔案並傳遞
  - Server Response
    - 若操作成功：回傳”<Response 200> [成功訊息]”
    - 若操作失敗：回傳”<Response 500> [錯誤訊息]”
-

# RESTful Service 功能操作 (Cont.)

---

31

## ■ 新增Table

- 不須手動新增Table
  - 直接新增資料至Table(上一頁PPT方法)
    - RESTful Service將自動偵測是否存在該Table
    - 若Table存在：Service將直接新增資料
    - 若Table不存在：Service將新增Table後再新增資料
-

# RESTful Service 功能操作 (Cont.)

32

## ■ 新增資料至Table或新增Table範例

➤ DB Name: **adas** ; Table Name: **car**

➤ POST <http://140.116.86.246:9999/adas/car>

➤ 欲傳送的**Json Data**:

- {"0": {"id": "0x00", "speed": "70"},  
"1": {"id": "0x11", "speed": "30"}}

➤ Python 程式碼

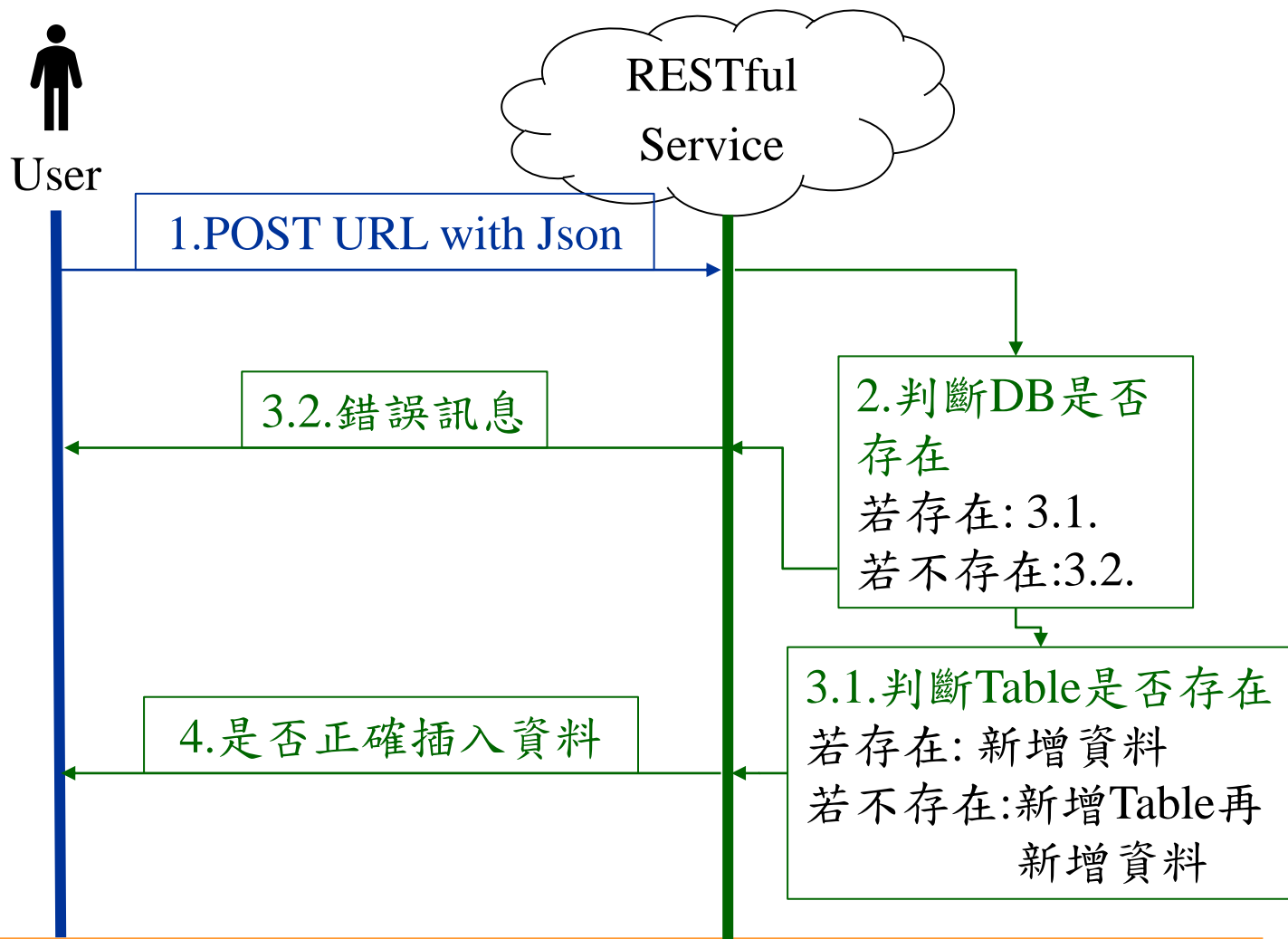
- `r = requests.post("http://140.116.86.246:9999/adas/car",  
                    json={"0": {"id": "0x00", "speed": "70"},  
                        "1": {"id": "0x11", "speed": "30"}}})`

```
>>> r = requests.post("http://140.116.86.246:9999/adas/car", json={"0": {"id": "0x00", "speed": "70"}, "1": {"id": "0x11", "speed": "30"}})
```

# RESTful Service 功能操作

33

## ■ Summery – 新增Table資料





# RESTful Service 功能操作 (Cont.)

34

## ■ 更新Table特定資料

※ RESTful Service不處理「Table資料唯一性」的問題

- 若Table中存在兩筆相同之欲更新資料  
=> 兩筆資料皆會被更新

➤ User必須指定(1)DB Name；(2)Table Name；(3)**Condition**

➤ 使用HTTP PUT

- URL: `http://140.116.86.246:9999/[DB Name]/[Table Name]/[Primary Key]=[Value]`

➤ 將欲更新之資料格式化為Json檔案並傳遞

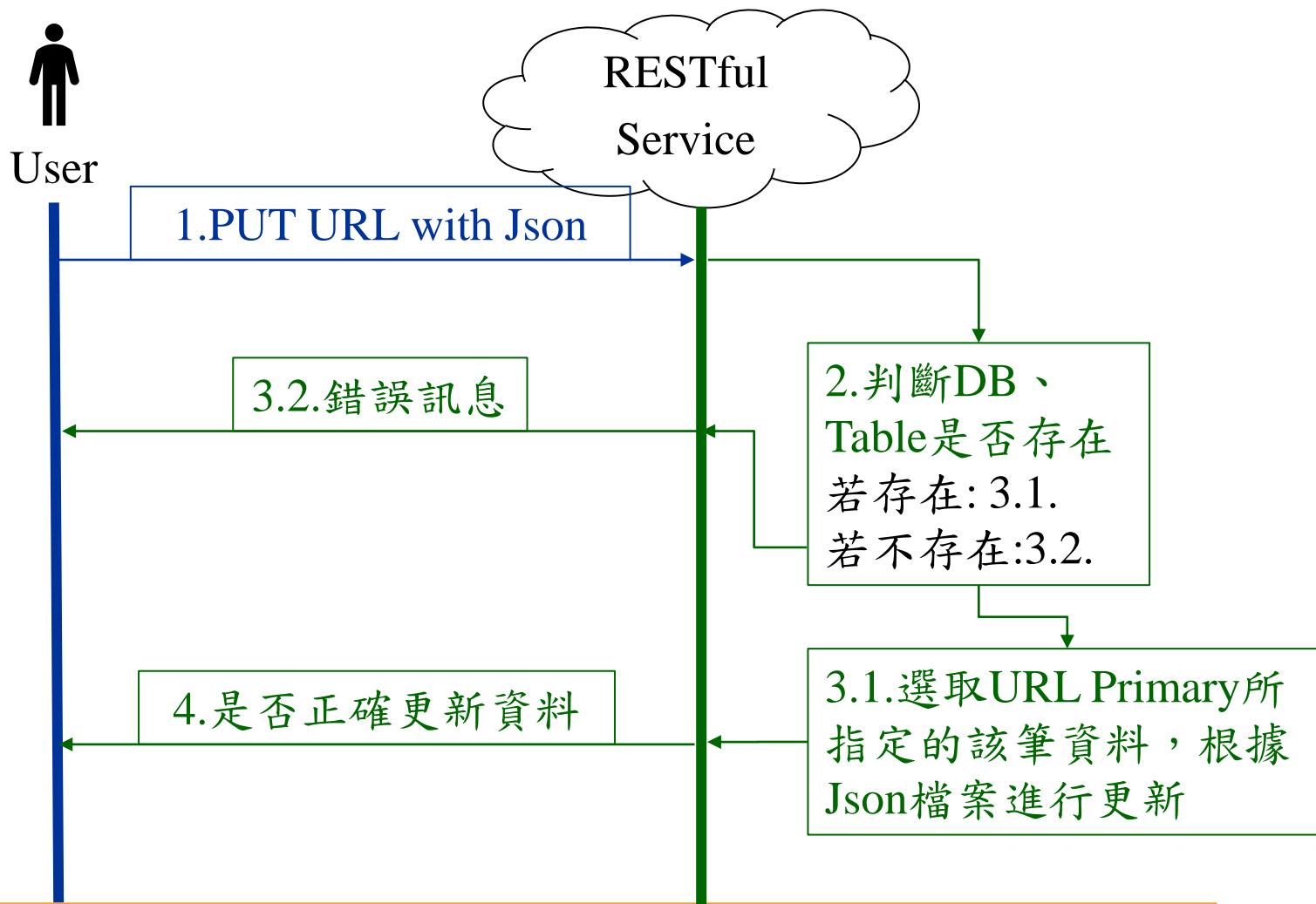
➤ Server Response

- 若操作**成功**：回傳”<Response **200**> [成功訊息]”
- 若操作**失敗**：回傳”<Response **500**> [錯誤訊息]”

# RESTful Service 功能操作

35

## ■ Summery – 更新Table特定資料



# Summary

---

36

- 透過URL指定資源
  - 透過HTTP協定(GET、POST、PUT)操作資源
  - 定義取得、新增、更新資料的Json檔案及其存入資料庫的格式
  - 取得、新增、更新資料的操作流程
-