



實驗一

Deadline: 12/4 11:59pm

LAB3-2 REST API

實驗目的

1. 學習如何呼叫 REST API

實驗環境

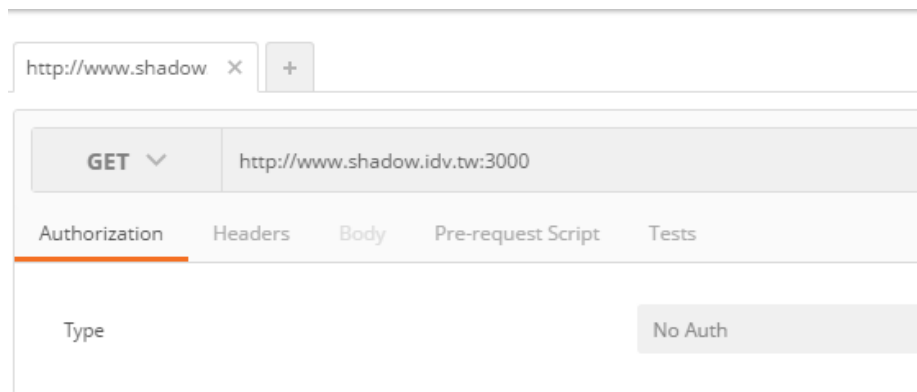
1. Chrome
2. Postman

實驗要求

1. 熟悉 REST API 的基本觀念
2. 熟悉 Postman 的基本操作
3. **不需**繳交任何東西，若已熟悉相關操作可以跳過

實驗步驟

1. 點參考[1]的連結下載 Chrome 的擴充功能 Postman
2. 找一組自己有興趣的 Service API，例如 Google 所提供的 Translate API、URL Shortener API 等。若沒找到可以使用助教用 Express [2]建立的簡單 REST API Server (<http://www.shadow.idv.tw:3000>)
3. 在網址列輸入目標的 URL，並在左邊選擇對應的 HTTP Request Type，這邊先以 GET 為例，如下圖。若為自己找的 API，可能會要求要有認證，請自行在 header 或是 authorization 分頁中加入對應的欄位。



4. 將 Request Type 改為 POST，並在 body 分頁輸入該服務要求的對應資料格式。若使用助教提供的 API，只限定使用 JSON 格式內容不限如下圖。



參考文件

1. Postman 下載點:
<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop>
2. ExpressJS 網站: <http://expressjs.com>



實驗二

Deadline: 12/4 11:59pm

LAB3-2 OpenStack API

實驗目的

1. OpenStack API 的基礎操作，利用 API 建立一台 Instance

實驗環境

1. Ubuntu 16.04 LTS
2. OpenStack (devstack allowed)

實驗要求

1. 利用任何語言(shell script、C/C++、Python、Javascript、.....)，寫出一個呼叫 OpenStack 的 REST API 建立 instance 的程式
2. 若要使用 Shell script，**禁止**直接使用 OpenStack 的 CLI API，請自行用 curl 這類工具呼叫 REST API

實驗步驟

1. 首先要準備一台 Network、Image、Flavor 都設定好的 OpenStack 環境，若 Lab 3-1 沒有成功架設起來可以使用 devstack 代替。
2. 要求 OpenStack 幫我們建立 instance 之前，我們必須先去 keystone 那邊要到一組 auth token，並將要到的 token 附在每個 request 的 header 裡面。首先發出要求 token 的 POST request 到 auth URL(通常是



controller:5000/v3/auth/tokens) ，附上對應的 request body 如下圖。

```
1 {
2   "auth": {
3     "identity": {
4       "methods": [
5         "password"
6       ],
7       "password": {
8         "user": {
9           "name": "admin",
10          "domain": {
11            "name": "Default"
12          },
13          "password": "123456"
14        }
15      }
16    },
17    "scope": {
18      "project": {
19        "name": "admin",
20        "domain": {
21          "name": "Default"
22        }
23      }
24    }
25  }
26 }
```

如果成功，得到的 response header 和 body 會像下圖所示。

```
1 {
2   "token": {
3     "issued_at": "2016-11-22T08:01:19.000000Z",
4     "audit_ids": [
5       "jBqJIjtStu0HaE1zFTtDmg"
6     ],
7     "methods": [
8       "password"
9     ],
10    "expires_at": "2016-11-22T09:01:19.000000Z",
11    "user": {
12      "domain": {
13        "id": "default",
14        "name": "Default"
15      },
16      "id": "eal1c04f5321c4f07a6bc36742a196af4",
17      "name": "admin"
18    }
19  }
20 }
```

| Body | Cookies | Headers (10) | Tests |
|---|---------|--------------|-------|
| Connection → Keep-Alive | | | |
| Content-Length → 283 | | | |
| Content-Type → application/json | | | |
| Date → Tue, 22 Nov 2016 08:01:17 GMT | | | |
| Keep-Alive → timeout=5, max=100 | | | |
| Server → Apache/2.4.18 (Ubuntu) | | | |
| Vary → X-Auth-Token | | | |
| X-Distribution → Ubuntu | | | |
| X-Subject-Token → gAAAAABYM_iPvLfdNgIV_CXpVXq7Ihoni7t96vcqRYWdch | | | |
| x-openstack-request-id → req-01394a30-e454-4177-bcec-7865f1b66f15 | | | |

而我們所需要的 token 便是藏在 header 中 X-Subject-Token 的欄位。

3. 拿著剛剛取得的 token，附在接下來所有 API call header 中的 X-Auth-

Token 欄位裡面。並依照 Compute API 的相關指示去建立一台

instance，分別是找到目標 flavor 的 ID、image 的 ID、以及建立

instance。這邊就不贅述，請大家自行參考文件[1]說明。

4. 本次 lab 要求安裝一台 image name 為 cirros、flavor name 為 m1.nano

的 instance，請各位務必使用 OpenStack 相關 API 查詢 image name 及



flavor name 的對應 ID，並利用對應的 ID 建立 instance。

5. 請將最後寫完的程式上傳，並附上相關編譯/執行方式在 Lab3-2.txt，

並壓縮為 Lab3-2.zip 上傳

參考文件

1. OpenStack API: <http://developer.openstack.org/api-guide/quick-start/index.html>

上傳檔案

- Lab3-2.zip