

第十九章

ANDROID網路程式

課程簡述

- 如何使用App呼叫Server API
- 如何處理JSON
- 如何建立自己的API Server
- 如何使用MongoDB
- 如何讓自己的API Server連接MongoDB並回傳結果
- 如何讓App呼叫自己的API Server

概念介紹

與SERVER溝通流程



API



API

- Application Programming Interface
- 程式撰寫時不同軟體模組間的接口
- Web API
 - `http://<主機位置>/<分類>/<分類>/...`
e.g.
`https://api.twitter.com/1.1/statuses/user_timeline.json`
- 最常使用GET或POST呼叫

API

- GET將參數帶在HTTP的Request網址中
 - 網址 `http://xxx.toright.com/api/?id=010101`
 - 封包

```
GET /?id=010101 HTTP/1.1
Host: xxx.toright.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-TW;
rv:1.9.2.13) Gecko/20101203 Firefox/3.6.13 GTB7.1 ( .NET CLR 3.5.30729)
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-tw,en-us;q=0.7,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: UTF-8,*
Keep-Alive: 115
Connection: keep-alive
```

API

- POST將參數帶在HTTP Request的封包中
 - 網址 `http://xxx.toright.com/api/insert`
 - 封包

```
POST / HTTP/1.1
Host: xxx.toright.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-TW;
rv:1.9.2.13) Gecko/20101203 Firefox/3.6.13 GTB7.1 ( .NET CLR
3.5.30729)
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-tw,en-us;q=0.7,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: UTF-8,*
Keep-Alive: 115
Connection: keep-alive

Content-Type: application/x-www-form-urlencoded
Content-Length: 9
id=010101
```


SERVER及資料庫



SERVER及資料庫

- 可以用PHP, ASP, ASP.NET, VB.NET, JSP來製作
- 資料庫常使用MySQL, SQL Server, Postgres SQL, Oracle等等
- 本次教學，用較新的**Node.js**為Server與**MongoDB**為資料庫

與SERVER溝通流程



JSON

- JavaScript Object Notation
- 輕量化資料交換格式

- Example:

https://graph.facebook.com/prettyklicks/posts?access_token=123975213079|nqKWO89vVW4QH_bNmKH-Wiy3W0w&limit=1

JSON OBJECT

- 以 { 開始，以 } 結束
- 內容 Key(索引值)與Value(內容)
- 內容可以是
 - 整數或浮點數、字串(要用""框起)、布林(true, false)、JSON Array、JSON Object
- 中間以冒號:分隔 { Key : Value }
- e.g.

```
{"name" : "android"}
```

JSON ARRAY

- 以 [開頭，以] 結果
- 中間內容有序列表，每個陣列物件以 , 分隔
- e.g.

```
{  
    "values": [  
        {"value", 1.0}, {"value", 2.0}, {"value", 3.0}  
    ]  
}
```

JSON範例

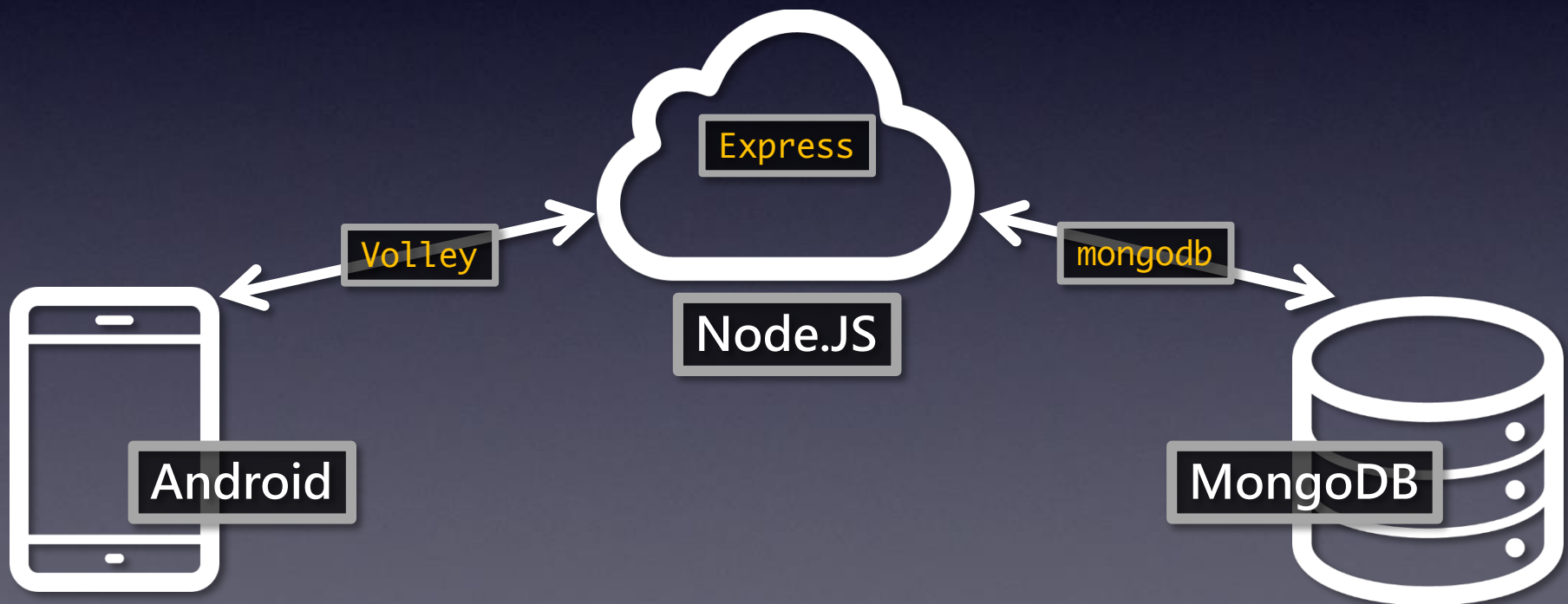
```
{  
  "id": "187393124625179",  
  "name": "Web Design"  
}
```

```
{  
  "id": "187393124625179",  
  "name": {"first" : "Justin", "last" : "Liu"}  
}
```

```
{  
  "id": "187393124625179",  
  "names": [  
    {"nickname", "John"},  
    {"nickname", "David"}  
  ]  
}
```

總結

- 即將使用的技術
 - 白字表示將會用到的平台
 - 橘字表示將會用到的函式庫或模組



ANDROID 呼叫 WEB API

APP呼叫WEB API



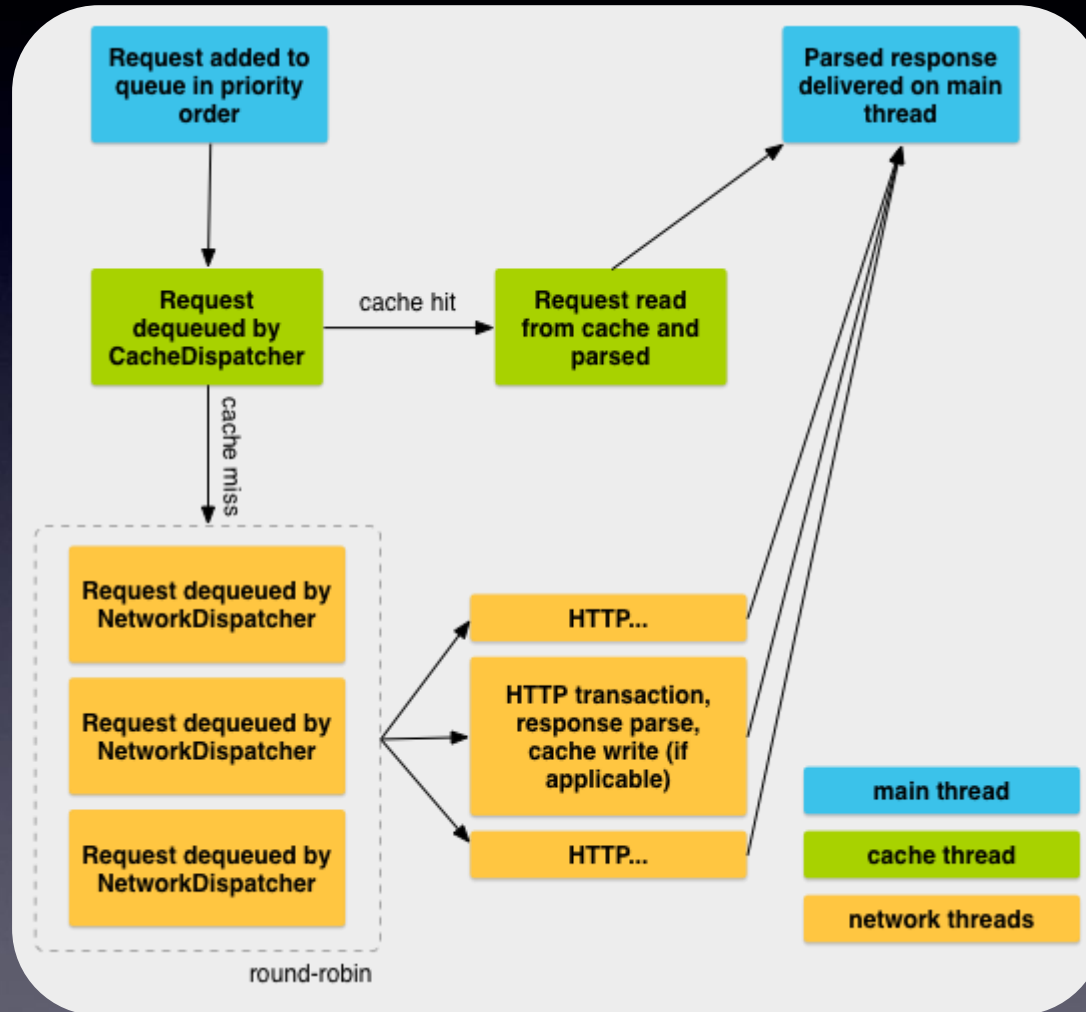
範例WEB API

- 牛刀小試
- GET <http://jsonplaceholder.typicode.com/posts/1>
 - 用瀏覽器打開試看看

ANDROID網路連線

- Android網路程式的困難
 - 不能在主執行緒進行網路連線
 - 多執行緒溝通的困難
 - 很多人不能了解AsyncTask來完成
 - 錯誤情況很難處理
- 使用第三方函式庫(Library) , VOLLEY

VOLLEY流程圖



VOLLEY的使用步驟

1. 設定使用網路的權限
2. 放入volley的函式庫
3. 建立Request
4. 使用NetworkManager
5. 接收回應與處理錯誤
6. 停止volley

1. 設定使用網路的權限

- 在AndroidManifest.xml中要設定Internet的 permission

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.networkcommunication"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-permission android:name="android.permission.INTERNET"/>
    .....
</manifest>
```

2. 放入VOLLEY的函式庫

- 將volley.jar放置到以下資料夾
<專案>/lib
- volley是開放原始碼專案，可由以下網址取得
 - <https://android.googlesource.com/platform/frameworks/volley>
- 更多volley的資訊
 - <http://developer.android.com/training/volley/index.html>

3. 建立REQUEST

- 在需要使用地方，加上Request

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        StringRequest request = new StringRequest(Request.Method.GET,  
            "http://jsonplaceholder.typicode.com/posts/1", mResponseListener,  
            mErrorListener);  
        NetworkManager.getInstance(this).request(null, request);  
    }  
}
```

3. 建立REQUEST

```
StringRequest request = new StringRequest(  
Request.Method.GET,  
"http://jsonplaceholder.typicode.com/posts/1",  
mResponseListener,  
mErrorListener);
```

- 參數1：GET或是POST(或是其他HTTP的方法)
- 參數2：API網址
- 參數3：Server回應後的Listener
- 參數4：錯誤出現時的Listener

4. 使用NETWORKMANAGER

- NetworkManager可以將Request藉由Volley發送給API Server

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        StringRequest request = new StringRequest(Request.Method.GET,  
            "http://jsonplaceholder.typicode.com/posts/1", mResponseListener,  
            mErrorListener);  
        NetworkManager.getInstance(this).request(null, request);  
    }  
}
```

5. 接收回應與處理錯誤

- 若連線沒有出錯且得到Server正常的回應，就會從Request的第三個參數得到結果
 - 由onResponse()收到Server回應的字串

```
private Listener<String> mResponseListener =  
new Listener<String>() {
```

```
    @Override
```

```
    public void onResponse(String string) {  
    }  
};
```

5. 接收回應與處理錯誤

- 若連線過程出錯，或Server回應錯誤，就會從Request的第四個參數回應
 - onResponse()會得到錯誤的通知

```
private ErrorListener mErrorListener = new ErrorListener() {  
  
    @Override  
    public void onResponse(VolleyError error) {  
    }  
};
```

6. 停止VOLLEY

- 最後在第一個使用NetworkManager的Activity，其中的onDestroy()呼叫stop()以便釋放資源

```
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    NetworkManager.getInstance(this).stop();  
}
```

剖析JSON與JSON ARRAY

剖析JSON

- 由外至內剖析，先取得JSONObject再藉由索引值取得對應的內容



剖析JSON

- 使用getInt(索引值)取得數字的值



```
json.getInt("userId")
```

```
{  
  "userId": 1,  
  "id": 1,  
  "title": "sunt ...",  
  "body": "quia et ..."  
}
```

剖析JSON

- 使用getInt(索引值)取得數字的值

```
{  
  "userId": 1,  
  "id": 1,  
  "title": "sunt ...",  
  "body": "quia et ..."  
}
```

json.getInt("id")

剖析JSON

- 使用getString(索引值)取得字串的值

```
{  
  "userId": 1,  
  "id": 1,  
  "title": "sunt ...",  
  "body": "quia et ..."  
}
```

`json.getString("title")`

剖析JSON

```
public void onResponse(String string) {  
    try {  
        JSONObject json = new JSONObject(string);  
        String title = json.getString("title");  
        String body = json.getString("body");  
  
        TextView text1 = (TextView) findViewById(R.id.textView1);  
        text1.setText(title);  
        TextView text2 = (TextView) findViewById(R.id.textView2);  
        text2.setText(body);  
    } catch (JSONException e) {  
        e.printStackTrace();  
    }  
}
```

剖析JSON ARRAY

- 試著將網址改為

<http://jsonplaceholder.typicode.com/todos>

```
[  
  {  
    "userId": 1,  
    "id": 1,  
    "title": "delectus aut autem",  
    "completed": false  
  },  
  {  
    "userId": 1,  
    "id": 2,  
    "title": "quis ut nam facilis et  
officia qui",  
    "completed": false  
  },  
  ...  
]
```

JSONArray ary

剖析JSON ARRAY

```
[  
  {  
    "userId": 1,  
    "id": 1,  
    "title": "delectus aut autem",  
    "completed": false  
  },  
  {  
    "userId": 1,  
    "id": 2,  
    "title": "quis ut nam facilis et  
officia qui",  
    "completed": false  
  },  
  ...  
]
```

`ary.getJSONObject(0)`

剖析JSON ARRAY

```
[  
  {  
    "userId": 1,  
    "id": 1,  
    "title": "delectus aut autem",  
    "completed": false  
  },  
  {  
    "userId": 1,  
    "id": 2,  
    "title": "quis ut nam facilis et  
officia qui",  
    "completed": false  
  },  
  ...  
]
```

`dry.getJSONObject(1)`

剖析JSON ARRAY

```
public void onResponse(String string) {
    try {
        JSONArray ary = new JSONArray(string);
        StringBuilder userIds = new StringBuilder();
        StringBuilder titles = new StringBuilder();
        for (int i = 0; i < ary.length(); i++) {
            JSONObject json = ary.getJSONObject(i);
            int userId = json.getInt("userId");
            userIds.append(userId);
            userIds.append(",");
            String title = json.getString("title");
            titles.append(title);
            titles.append(",");
        }
        TextView text1 = (TextView) findViewById(R.id.textView1);
        text1.setText(userIds.toString());
        TextView text2 = (TextView) findViewById(R.id.textView2);
        text2.setText(titles.toString());
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```


作業1

- 使用GET方式取得Flickr的圖片資訊
 - URL:
`http://www.flickr.com/services/feeds/photos_public.gne?tags=soccer&format=json&jsoncallback=?`
- 將抓回圖片資訊，標題呈現在ListView上
- 點下ListView，會顯示AlertDialog
 - AlertDialog標題呈現圖片標題
 - AlertDialog的Message呈現關於圖片說明
 - AlertDialog的按鈕按下後要用瀏覽器開啟圖片資訊的網頁
 - AlertDialog的另一個按鈕按下後關閉AlertDialog

作業1

