

JSONP 介紹

written Jul 6th, 2015 | in [jquery](#) | [comments](#)

JSONP(JSON with Padding), 聽起來跟JSON很像? 兩者有什麼關連嗎?! JSONP是一種跨網域資料交換的**方式**, 而JSON則是一種資料交換的**格式**。而兩者的關聯就是JSON是AJAX在交換資料所常用的格式, 而JSONP則是AJAX為突破同源政策(Same-origin policy), 而可讓不同網域之間一樣可以靠xhr取得資料的手段。

JSONP由來:

在探討JSONP前一樣先來看看它是為何而生的呢? 因為AJAX基於安全的考量, 不允許對不同網域的網站發送請求, 以防某些網站會在背後偷偷試圖連到其他網站去幹壞事(如: 銀行網站之類的); 但跨域的AJAX需求仍然常發生, 因此當然有諸多因應之道, 常見的有下:

- 在同樣的網域下建立一支proxy api再例用這proxy去call外部的資源
- 伺服器端必須Access-Control-Allow-Origin設為*或給某些特定網域
- 利用script tag的src載入

而JSONP主要則是利用其中的第三點來達成, 這是因為src這個attribute不受是否跨網域的限制(e.g. img, script, iframe), 所以我們可以利用這個特性傳回一段**可執行**的js code, 例如

對於某個外部請求, server會回傳一段js code

```
1 alert('WTF')
```

而client這邊可以這樣來獲取

```
1 <script src="http://somewhere.com/test"></script>
```

結果就是會在頁面上跳出一句WTF, 由此證明我們可以利用src不受跨網域限制的特性來克服以往AJAX所達不到的目的, 而我們更進一步可以讓server端將希望回傳的資料(JSON)包成可執行的js code回傳給我們 如:

```
1 someFunction({ foo: 'bar', sns: 'yoona'})
```

而我們只需要在client裡面先定義好someFunction這個function就可以發揮如同callback的效用達成跨網域的請求了!

JSONP目的

所以, JSONP主要就是讓server端將所有要回傳的資料包在任意名稱的function參數裡回傳給client端使用, JSON with Padding的名稱也非常貼切它的目的。而剛剛所提到任意名稱的function, 是因為我們既然要事先定義該function, 才可使用server回傳給我們的js code, 那必然也得讓server要傳什

麼名稱給我們，簡單的說就是需要動態的產生function名，而這只需要把function的名稱當做參數傳過去即可，而後端的處理可以這樣寫：

some_controller.rb

```
1  def jsonp
2    @data = { name: 'Yoda', role: 'master' }.to_json
3    @func_name = params[:callback]
4  end
```

jsonp.js.erb

```
1  <%= @func_name %>(<%= raw @data %>)
```

client的code會長這樣

```
1  <script>
2    var say_hi = function(data){
3      alert('Hi ' + data.name)
4    }
5  </script>
6  <script src="http://somewhere.com/test?callback=say_hi">
7  </script>
```

JSONP 使用 jQuery

標準用法

實際上JSONP不算是官方正式提出用來解決跨網域需求存取的解法，算是非官方的協定。然而JSONP本身也非使用xhr物件來達到跨網域的請求，但jQuery還是支援jsonp而且用法相當簡單：

```
1  function test(data){
2    //do_something
3  }
4
5  $.ajax({
6    url: 'http://somewhere.com/test',
7    type: 'get',
8    dataType: 'jsonp',
9    jsonp: 'callback',
10   jsonpCallback: 'test'
11 })
```

只要把dataType設為jsonp即可。而參數jsonp: 預設為callback，所以是callback=?，如果jsonp設為foo那就是 foo=?。

jsonpCallback: 預設是由jQuery亂數產生的唯一值當做function name，如果 jsonpCallback設為bar，那就是 callback=bar。最後前端就必須要有定義function bar讓他可以執行。

建議好像是讓兩者都為預設就可以了，因為cache的問題，亂數產生的function name像是callback=XXXXX就會每次都被視為新的request就比較不會有被cache住，而拿不到即時更新過資料的問題了。

精簡用法

```
1  $.ajax({
2    url: 'http://somewhere.com/test',
3    type: 'get',
4    dataType: 'jsonp',
5    success: function(data){
6      //do_something
7    },
8    error: function(){
9      console.log('fail')
10   }
11  })
```

jsonp預設為callback，所以可以不需要寫，jsonpCallback也建議不用設定，使用預設的亂數值即可，而且聰明的jQuery會自動把回傳的data帶入success這個callback使用，也不需要額外定義function去接了，非常方便！！

網路上看到使用\$.getJSON網路上有可以這樣試，jQuery似乎會自動去偵測callback這個querystring(不過本人還沒測試過)

```
1  $.getJSON('http://somewhere.com/test?callback=?', function(data){
2    //do_something
3  })
```

以上就是關於jsonp的學習心得與分享，如有錯誤或其他建議都歡迎指正～
