Django CORS跨域資源共享

如果你使用公開的IP啟動你的網站，你會發現:Ajax網頁輸出結果沒有了!

按下F12會出現紅色CORS之類的訊息。你需要設定CORS!

什麼是CORS？

CORS(跨域資源共享，Cross-Origin Resource Sharing)是一種跨域訪問的機制，可以讓Ajax實現跨域訪問。

API部署在DomainA上

Ajax網頁部署在DomainB上，Ajax網頁會向DomainA上API發送請求，返回數據。

某使用者位於DomainC，使用者從DomainC訪問DomainB的Ajax網頁，請求數據，此過程就發生了跨域訪問。

domain C🡪 domain B🡪 domain A

是不是跨了3個網域?

此時使用Ajax來請求API就會失敗。

Chrome提示：

No 'Access-Control-Allow-Origin' header is present on the requested resource.

我們必須對API進行一些特別的設計，就是允許CORS(跨域資源共享)。

這就是Restful API

讓Django伺服器提供CORS標頭給你的網頁

安裝套件

pip install django-cors-headers

(1) 第1處修改

INSTALLED\_APPS = (

...

'corsheaders', #跨站請求

...

)

(2) 第2處修改

Insert the following line before ‘….CommonMiddleware’

MIDDLEWARE\_CLASSES = (

...

'corsheaders.middleware.CorsMiddleware', #跨站請求

'django.middleware.common.CommonMiddleware',

...

)

(3) 第3處修改: 在Setting.py最後面加上這一行

# 允許所有跨站請求  
CORS\_ORIGIN\_ALLOW\_ALL = True

若沒有設定允許所有跨站請求，瀏覽器console會有以下錯誤

Access to XMLHttpRequest at 'http://163.18.23.120:8000/api\_get\_sentiment/' from origin 'http://127.0.0.1:8001' has been blocked by CORS policy: Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource.

(4) 第4處修改:

SECURE\_CROSS\_ORIGIN\_OPENER\_POLICY = 'None'

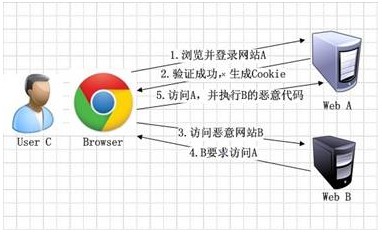
否則瀏覽器會出現一下訊息:

Unless set to None, the SecurityMiddleware sets the Cross-Origin Opener Policy header on all responses that do not already have it to the value provided.

CSRF(Cross Site Request Forgery)跨站請求偽造

認識CSRF漏洞

CSRF 是一種 Web 上的攻擊手法，全稱是 Cross Site Request Forgery，跨站請求偽造



[圖片](https://www.jianshu.com/p/a178f08d9389)來源

Django內定啟動跨站請求偽造（Cross-site Request Forgery，CSRF）防護機制。

若沒有關閉，則以第三方軟體呼叫API，會出現以下錯誤訊息!

Forbidden!

CSRF verification failed. Request aborted.

因為是API程式，我們必須將CSRF關閉，只針對某個API程式關閉CSRF，方式如下:

from django.views.decorators.csrf import csrf\_exempt

@csrf\_exempt

def test\_api( request ):

為甚麼RESTful API必須關閉CSRF防護?

API就是開放給第三方程式呼叫讀取資料之用，有跨站的需要。

對於讀取資料的服務，無安全之虞，不需要進行CSRF 的保護。

例如:

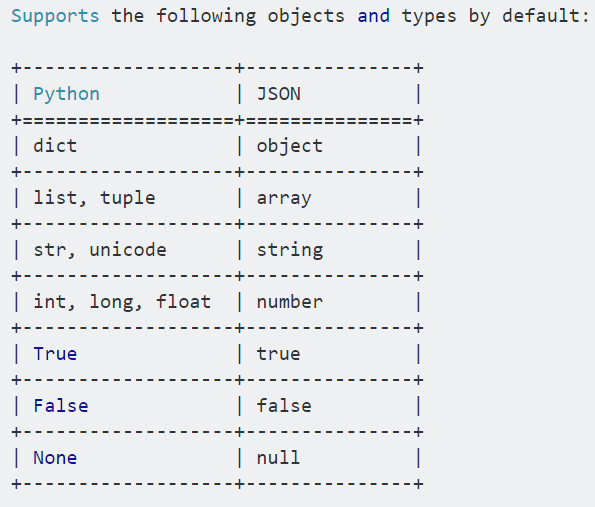
銀行系統中轉帳的請求會直接改變帳戶的金額，會遭到 CSRF 攻擊，需要保護。而查詢餘額是對金額的讀取操作，不會改變資料，CSRF 攻擊無法解析伺服器返回的結果，無需保護。

資料來源

An API pretty much precludes CSRF, because its entire purpose is generally to allow 3rd-party entities to access and manipulate data on your site (the "cross-site" in CSRF). So, yes, I think as a rule any API view should be CSRF exempt. However, you should still follow best practices and protect every API-endpoint that actually makes a change with some form of authentication, such as OAuth.

API程式為何對於資料的傳送使用json格式?

JSON encoding is supported for the following native python types out of the box.



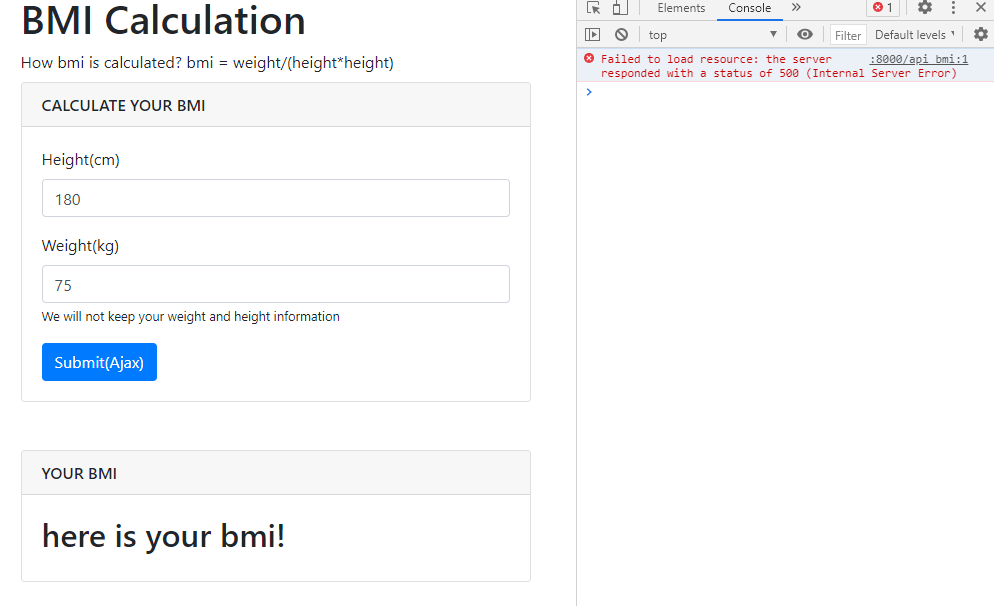
views.py函數中python數字、字串、list可以直接都給jsonResponse沒問題。

但是，

views.py函數中的dict, numpy等會被視為物件，必須進行以下處理方式如下:

* dict字典要用json.dumps()處理轉成字串str格式之後，再用JsonResponse()送出。(因為在網頁中會被轉成字串後其格式不對)
* pandas的欄位若為數字(文字沒問題): Pandas的欄位若為數字，則其格式是numpy格式，不能傳給網頁、或是給jsonResponse使用。必須預先轉成一般的int, float等格式。
* 中文字會呈現亂碼解決方式: json.dumps(sentiment\_prob, ensure\_ascii=False)

Trouble shooting



Internal Server Error

urls.py

|  |
| --- |
| urlpatterns = [  path('', views.home),  path('api\_bmi/', views.api\_get\_bmi),  ]  Notice the URL paths |

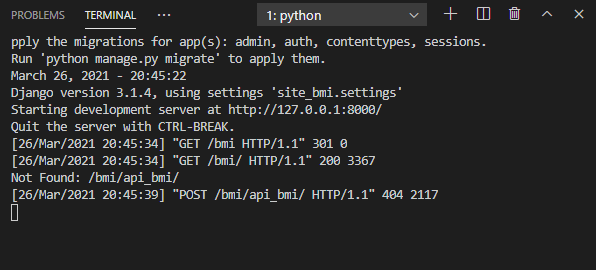
home.html

|  |
| --- |
| jQuery.ajax({  type: "POST",  url: "http://127.0.0.1:8000/api\_bmi/",  //url: "/api\_bmi/", //This is OK.  data: {  "height": height,  "weight": weight,  }, //pass to server  ……  });//ajax |

Error Message:

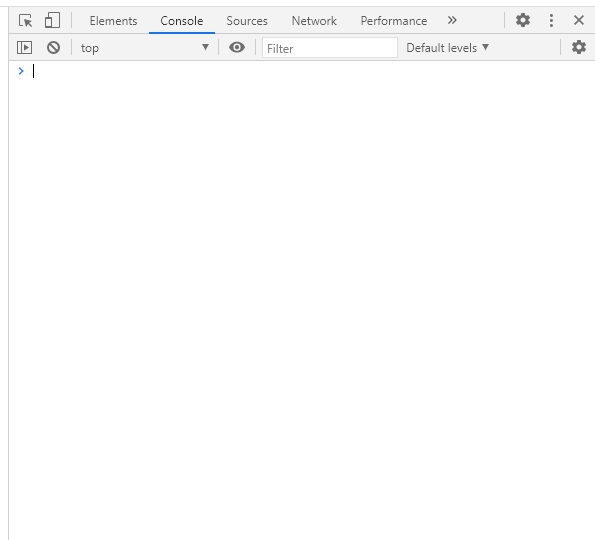
RuntimeError: You called this URL via POST, but the URL doesn't end in a slash and you have APPEND\_SLASH set. Django can't redirect to the slash URL while maintaining POST data. Change your form to point to 127.0.0.1:8000/api\_bmi/ (note the trailing slash), or set APPEND\_SLASH=False in your Django settings.

Not Found: /bmi/api\_bmi/



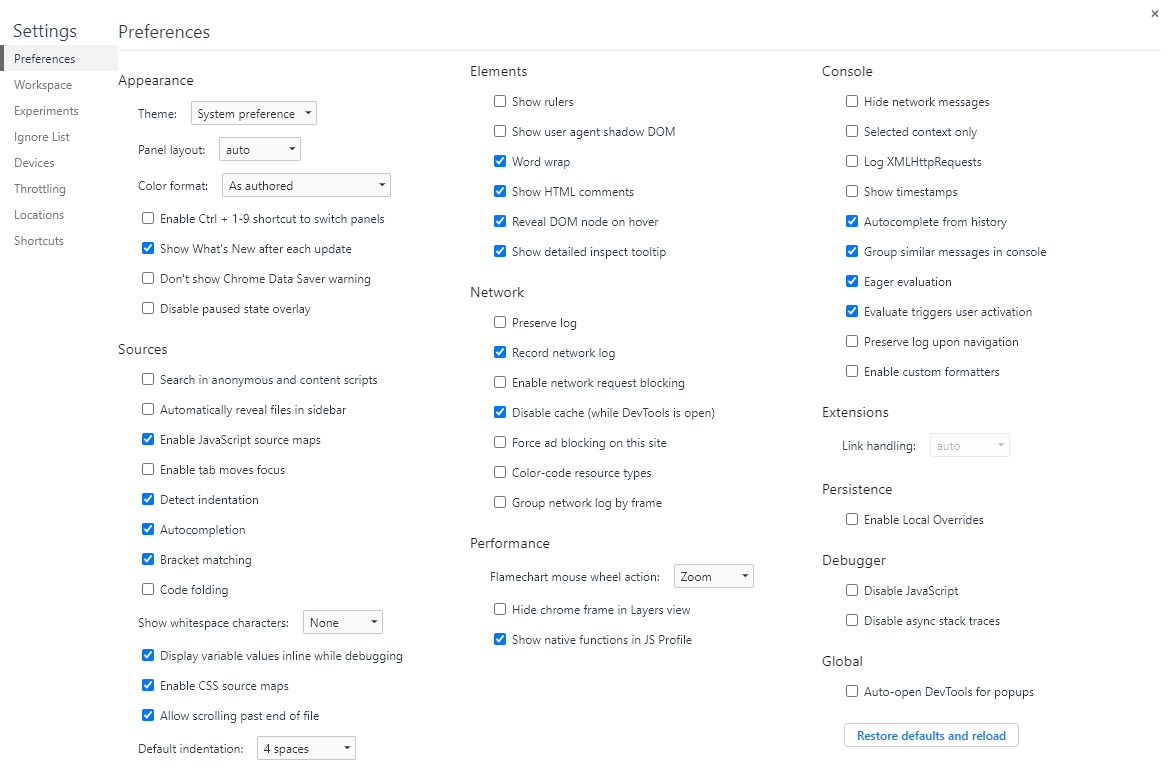
If everything is fine, but it still can’t work (e.g., “Not Found: /bmi/api\_bmi/”), what can you do?

1. 清除Chrome瀏覽器快取資料
2. Disable cache不要快取



Click it.

按一下這裡設定



Check it.

Disable cache

不要快取