

Network Programming

- 網路程式設計 -

Cookies 與 Sessions

授課教師：張珀銀 老師

06

Cookies 與 Sessions

6-1 關於 Cookie 與 Session

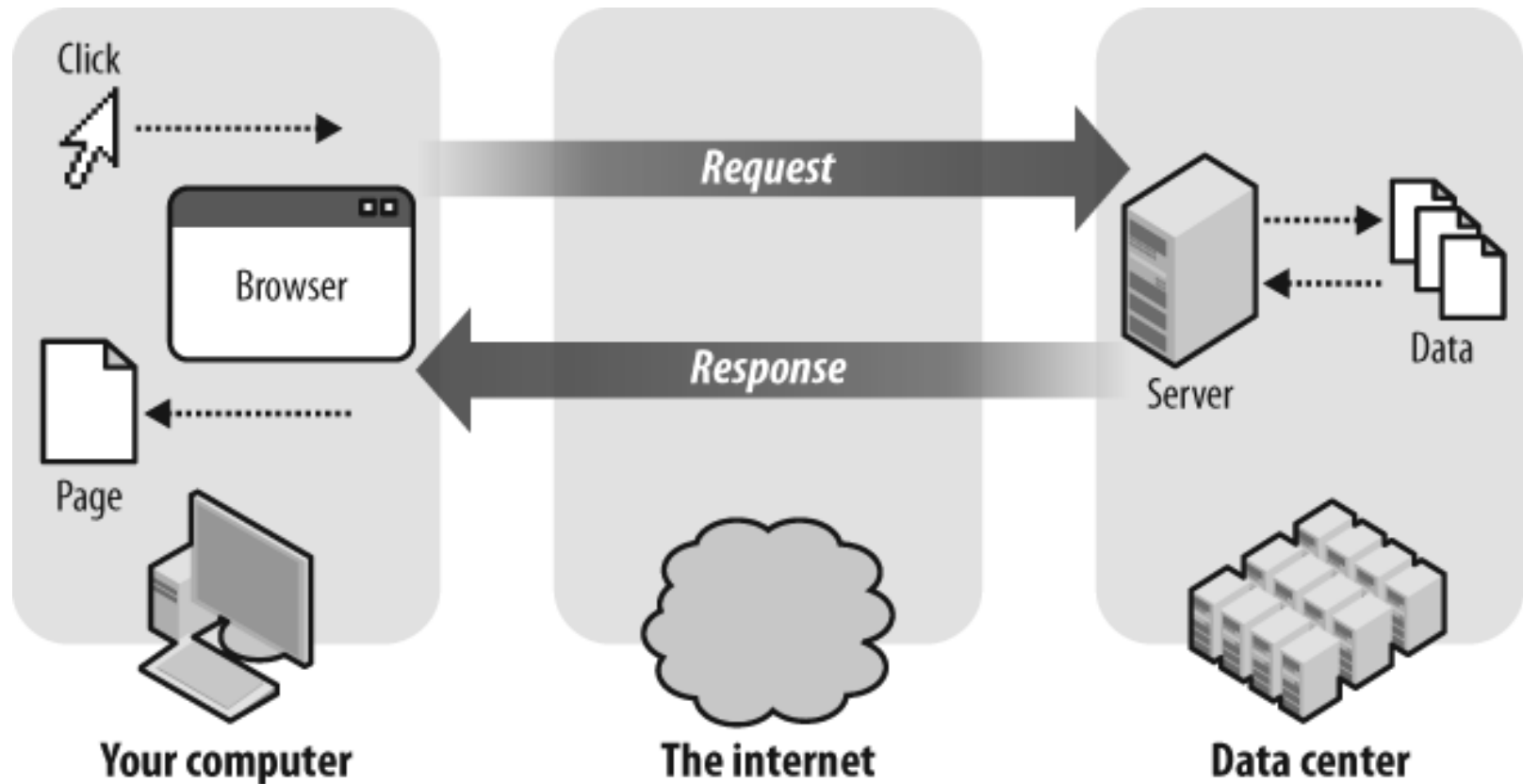
6-2 Cookie 的使用

6-3 Session 的使用

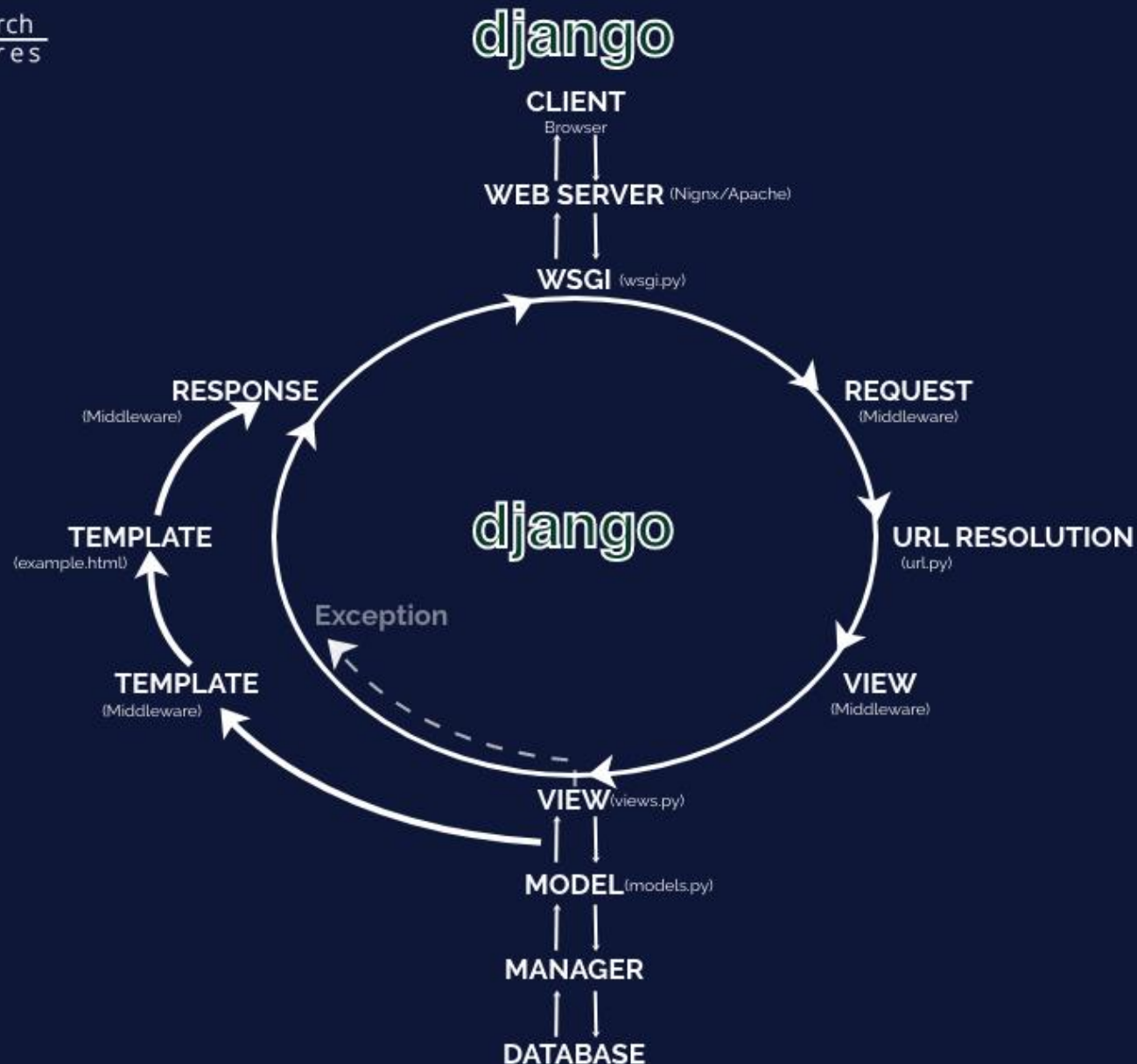


前一單元內容回顧

Django Request and Response Lifecycle



Django Request and Response Lifecycle



settings.py × urls.py × views.py ×

```
32 'django.contrib.admin',
33 'django.contrib.auth',
34 'django.contrib.contenttypes',
35 'django.contrib.sessions',
36 'django.contrib.messages',
37 'django.contrib.staticfiles',
38 'CookieSessionApp',
39 ]
```

```
40
41 MIDDLEWARE = [
42     'django.middleware.security.SecurityMiddleware',
43     'django.contrib.sessions.middleware.SessionMiddleware',
44     'django.middleware.common.CommonMiddleware',
45     'django.middleware.csrf.CsrfViewMiddleware',
46     'django.contrib.auth.middleware.AuthenticationMiddleware',
47     'django.contrib.messages.middleware.MessageMiddleware',
48     'django.middleware.clickjacking.XFrameOptionsMiddleware',
49 ]
```



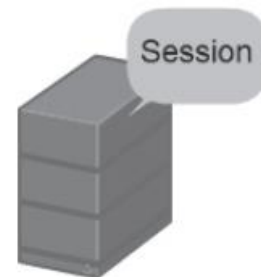
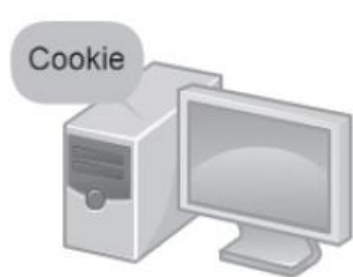
1

關於 Cookie 與 Session

6-1 關於 Cookie 與 Session

6.1.1 為什麼要使用Cookie與Session？

Cookie 與Session 的存在就是為了解決網站不能保存狀態的問題。以一般網站最常見的會員系統來說，當會員以帳號密碼登入系統的同時，程式可以有二個方式來記住登入會員的資料：一個方法是在登入者的電腦中放入一個小檔案來記憶，這個就是Cookie；另一個方法是在伺服器的記憶體產生一個空間來記憶，這個就是Session。



- ▲ 為了維持使用者的狀態，可以使用用戶端的 **Cookie** 或是伺服器端的 **Session** 來幫忙。

6.1.2 Cookie與Session的比較

關於 Cookie

因為Cookie 是以檔案的型態儲存在瀏覽者的電腦，有以下限制：

1. 每個使用者的瀏覽器只能儲存使用300 個Cookie 。
2. 每個瀏覽器只能針對同一個伺服器存取20 個Cookie 。
3. 每個Cookie 的大小最多僅4k Bytes 的容量。
4. 瀏覽器可以設定關掉Cookie 的功能，如此可能會造成Cookie 無法使用。

關於 Session

Session 會將使用者資訊儲存在伺服器端暫存檔中，當瀏覽者進入網站伺服器瀏覽時，在Session 的開啟狀態下即會開始記錄使用者所賦予的資訊，一直到Session的有效期限結束或刪除Session 才結束Session 的使用。

在程式運作正常的考量下，Session 因為存在伺服器端，即使用戶端的瀏覽器關閉Cookie 的使用時，Session 仍可正常運作。



2

Cookie 的使用

6-2 Cookie 的使用

6.2.1 存取Cookie 資料

允許儲存及讀取Cookie

在Chrome 瀏覽器的「設定」的「隱私權和安全性」可設定儲存及讀取Cookie 資料的狀況：建議使用預設的「在無痕模式中封鎖第三方 Cookie」。



儲存Cookie 資料

```
set_cookie(key,value='',max_age=None,expires=None)
```

參數	說明
key	Cookie 儲存時的名稱。
value	Cookie 儲存值並以 URL 編碼。因為是以明碼的方式儲存，所以不建議儲存重要資料。省略時預設值為空字串。
max_age	Cookie 的持續時間，單位為秒。省略時預設為瀏覽器開啟時間，當瀏覽器關閉後 Cookie 即自動刪除。
expires	到期時間。expires 是一個 UTC "Wdy, DD-Mon-YY HH:MM:SS GMT" 格式的字符串，或是一個 datetime.datetime 物件。

例如：以HttpResponse 建立response 物件，再以response 物件的set_cookie 方法儲存Cookie 。

```
from django.http import HttpResponse  
response = HttpResponse('TestCookie')  
response.set_cookie("TestCookie", " 這是 Cookie 的內容 ")
```

若要設定 Cookie 的有效持續時間為 1 個小時(3600 秒) 方式為：

```
response.set_cookie("TestCookie", " 這是 Cookie 的內容 ",max_age=3600)
```

讀取Cookie 的資料

可以利用COOKIES 函式以字典的方式讀出Cookie 中儲存的資料，其格式如下：

```
request.COOKIES[ 名稱 ]
```

範例：儲存並顯示Cookie 值

以下範例示範如何儲存並顯示Cookie 值。

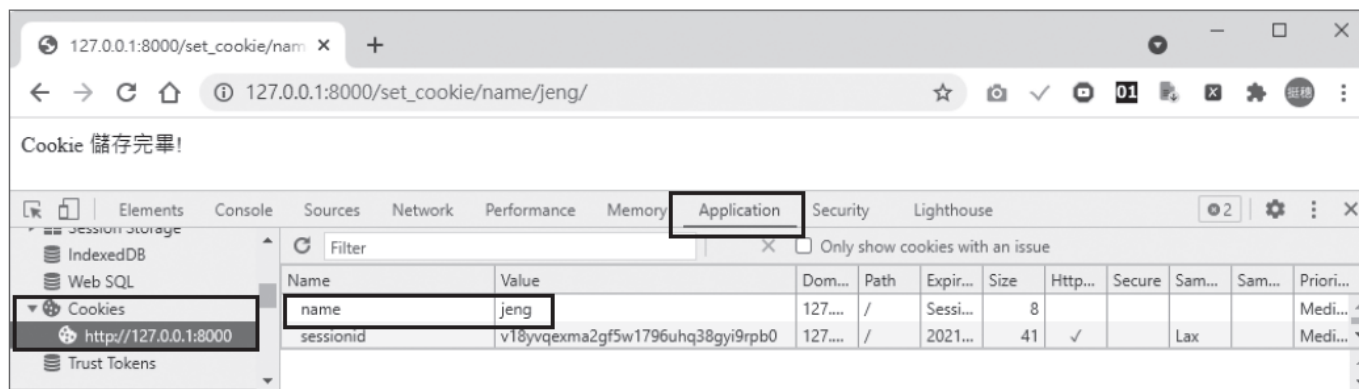
URL 定義「127.0.0.1:8000/set_cookie/<str:key>/<str:value>/」和
「127.0.0.1:8000/get_cookie/<str:key>/」網址。

CookieSession\urls.py

```
urlpatterns = [  
    path('set_cookie/<str:key>/<str:value>/', views.set_cookie),  
    path('get_cookie/<str:key>/', views.get_cookie),  
]
```

使用Google Chrome 瀏覽器，輸入

「http://127.0.0.1:8000/set_cookie/name/jeng/」儲存 Cookie，儲存後在頁面上按右鍵，在右鍵功能中按 檢查，選擇 Application\ Cookies 中的「<http://127.0.0.1:8000>」即可看到儲存的Cookie。它的名稱是「name」，內容是「jeng」。



輸入「http://127.0.0.1:8000/get_cookie/name/」可取得名稱為「name」的 Cookie內容。



定義 set_cookie() 和 get_cookie() 自訂函式。

CookieSessionApp\views.py

```
1  from django.shortcuts import render
2  from django.http import HttpResponseRedirect
3  import datetime
4
5  def set_cookie(request, key=None, value=None):
6      response = HttpResponseRedirect('Cookie 儲存完畢!')
7      response.set_cookie(key, value)
8      return response
9
10 def get_cookie(request, key=None):
11     if key in request.COOKIES:
12         return HttpResponseRedirect('%s : %s' %(key, request.COOKIES[key]))
13     else:
14         return HttpResponseRedirect('Cookie 不存在!')
```

Cookie 字典

Cookie 的格式是字典，因此可以迴圈方式從字典的items() 項目中，取得字典的key和value。

例如：定義一個get_allcookies() 自訂程序取得Cookie 中所有名稱和內容，存取前最好以「if request.COOKIES != None」檢查Cookie 是否存在。

```
def get_allcookies(request):  
    if request.COOKIES != None:  
        strcookies=""  
        for key1,value1 in request.COOKIES.items():  
            strcookies= strcookies + key1 + ":" + value1 + "<br>"  
        return HttpResponse('%s' %(strcookies))  
    else:  
        return HttpResponse('Cookie 不存在!')
```

6.2.2 Cookie 的有效時間

設定 Cookie 的持續時間

```
response.set_cookie("TestCookie", " 這是 Cookie 的內容 ",max_age=3600)
```

設定 Cookie 的到期時間

例如：設定到期時間為明日「0:0:0」，這就必須使用`expires` 參數，其中`tomorrow`是以現在時間加 1 日，也就是到明天，然後以`replace` 重設時、分、秒為「0:0:0」，最後再利用`strftime` 將日期換轉成指定的格式。

```
tomorrow = datetime.datetime.now() + datetime.timedelta(days = 1)
tomorrow = datetime.datetime.replace(tomorrow, hour=0, minute=0, second=0)
expires = datetime.datetime.strftime(tomorrow, "%a, %d-%b-%Y %H:%M:%S GMT")
response.set_cookie("counter",counter,expires=expires)
```

6.2.3 刪除 Cookie

```
delete_cookie[ 名稱 ]
```

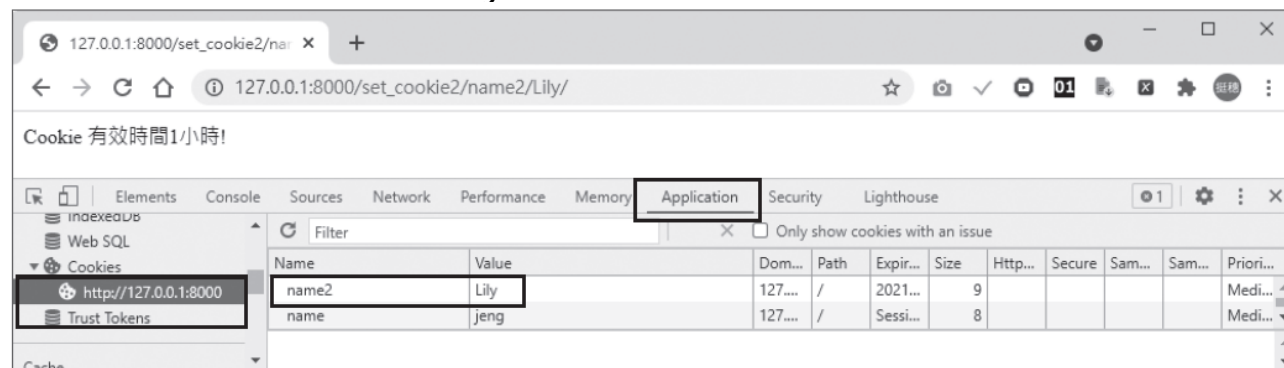
例如：刪除名稱為TestCookie 的Cookie。

```
from django.http import HttpResponse
response = HttpResponse('Delete Cookie')
response.delete_cookie("TestCookie")
```

範例：設定Cookie 持續時間

以下範例將示範如何設定Cookie 持續時間，也可以刪除指定的Cookie。

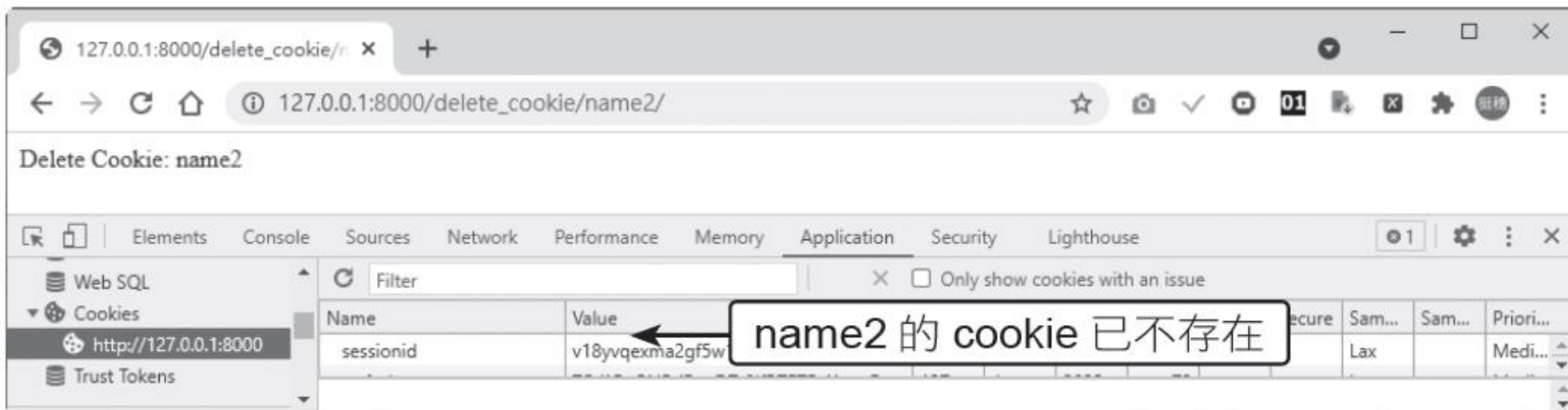
輸入「http://127.0.0.1:8000/set_cookie2/name2/Lily/」儲存Cookie，它的名稱是「name2」，內容是「Lily」，Cookie 持續時間為1 小時。



關閉瀏覽器後再重新啟動，輸入「http://127.0.0.1:8000/get_cookie/name2/」。
因為Cookie 持續時間為1 小時，因此可取得名稱為「name2」的Cookie 內容。



輸入「http://127.0.0.1:8000/delete_cookie/name2/」則可以立即刪除名稱為「name2」的Cookie 內容。



定義set_cookie2() 和delete_cookie() 自訂函式。

CookieSessionApp\views.py

```
16 def set_cookie2(request, key=None, value=None):
17     response = HttpResponse('Cookie 有效時間 1 小時!')
18     response.set_cookie(key, value, max_age=3600)
19     return response
20
21 def delete_cookie(request, key=None):
22     if key in request.COOKIES:
23         response = HttpResponse('Delete Cookie: '+key)
24         response.delete_cookie(key)
25         return response
26     else:
27         return HttpResponse('No cookies: ' + key)
```

範例：顯示使用者今天瀏覽本頁面的次數

以下將利用Cookie 來記錄使用者今天瀏覽本頁的次數，若是過了今天則記錄的次數歸零，重新計算。

URL 定義「127.0.0.1:8000/」和「127.0.0.1:8000/index/」網址。

```
CookieSession\urls.py
urlpatterns = [
    path('', views.index),
    path('index/', views.index),
]
```

輸入「http://127.0.0.1:8000/」或「http://127.0.0.1:8000/index/」會顯示瀏覽本頁的次數，網頁重整或關閉瀏覽器後重新開啟，瀏覽的次數會由1 開始累增。



A yellow circle with a black number 3 inside, set against a background of blue and yellow geometric shapes.

3

Session 的使用

6-3 Session 的使用

6.3.1 Session 的運作原理

當使用者使用瀏覽器連線到伺服器網站時，網站伺服器會自動派發一個SessionID 給這次的連線動作，網站程式即可依照這個SessionID 分辨使用者來處理所儲存的狀態。事實上：在預設的狀態下，Session 會將伺服器所派發的SessionID 加密處理後以Cookie 的方式儲存在用戶端來記錄狀態，同一個網站的不同網頁可以藉由這個Cookie 的記錄來維持同一個SessionID 的狀態。

6.3.2 安裝 Session App

CookieSession\settings.py

```
33  INSTALLED_APPS = [  
34      'django.contrib.admin',  
35      'django.contrib.auth',  
36      'django.contrib.contenttypes',  
37      'django.contrib.sessions', # 自動安裝 Session 的 app  
38      'django.contrib.messages',  
39      'django.contrib.staticfiles',  
40      'CookieSessionApp', # 新增的 app  
41  ]  
42  
43  MIDDLEWARE = [  
44      'django.middleware.security.SecurityMiddleware',  
45      'django.contrib.sessions.middleware.SessionMiddleware',  
46      # 自動安裝 Session 的 app  
47      'django.middleware.common.CommonMiddleware',  
48      'django.middleware.csrf.CsrfViewMiddleware',  
49      'django.contrib.auth.middleware.AuthenticationMiddleware',  
50      'django.contrib.messages.middleware.MessageMiddleware',  
51      'django.middleware.clickjacking.XFrameOptionsMiddleware',  
52  ]
```

6.3.3 存取 Session 資料

儲存session 資料

以字典方式可以設定session 的值，其格式如下：

```
request.session[ 名稱 ] = 值
```

讀取 session 資料

Session 儲存之後，我們就可以該鍵名稱取得Session 資料，其格式如下：

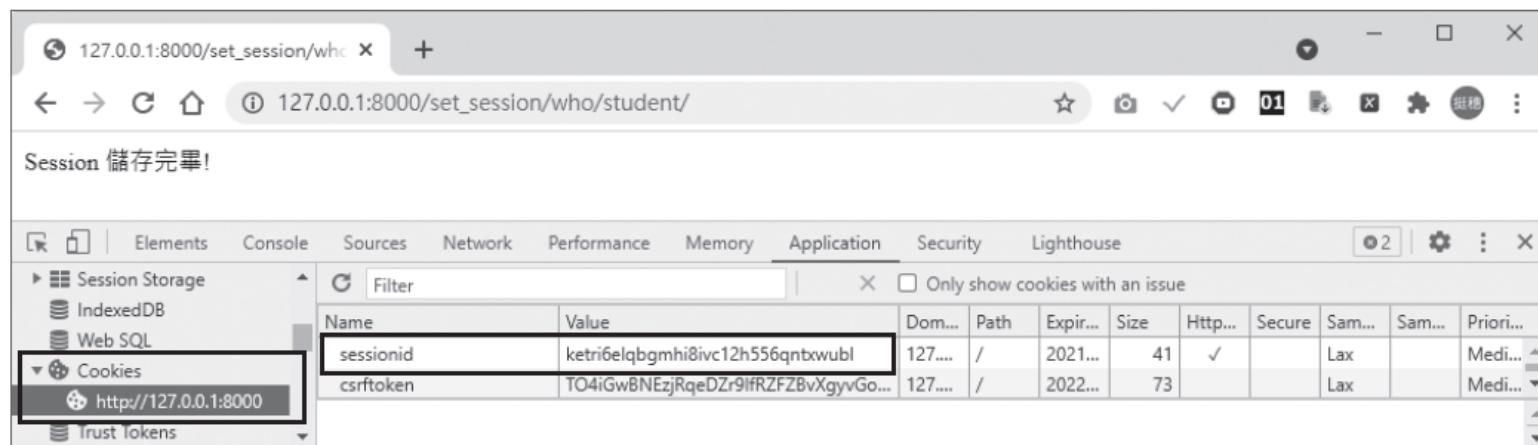
```
變數 = request.session[ 名稱 ]
```

範例：儲存並顯示Session 值

以下的範例將示範如何儲存並顯示Session 值。

輸入「`http://127.0.0.1:8000/set_session/who/student/`」儲存Session，名稱為「who」，內容為「student」，這個Session 會儲存在伺服器端。

儲存後在頁面上按右鍵，在右鍵功能中按 **檢查**，選擇**Application \ Cookies** 中的「`http://127.0.0.1:8000`」，可看到Session 將伺服器所派發的SessionID 加密處後以Cookie 的方式儲存在用戶端，它的名稱是「sessionid」，內容是一個編碼過的字串。



輸入「`http://127.0.0.1:8000/get_session/who/`」即可取得名稱為「who」的 Session 內容。



定義 `set_session()` 和 `get_session()` 自訂函式。

CookieSessionApp\views.py

略……

```
51 def set_session(request, key=None, value=None):
52     response = HttpResponse('Session 儲存完畢!')
53     request.session[key]=value
54     return response
55
56 def get_session(request, key=None):
57     if key in request.session:
58         return HttpResponse('%s : %s' %(key, request.session[key]))
59     else:
60         return HttpResponse('Session 不存在!')
```

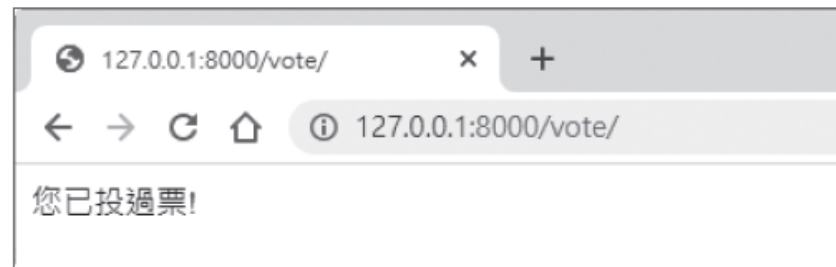
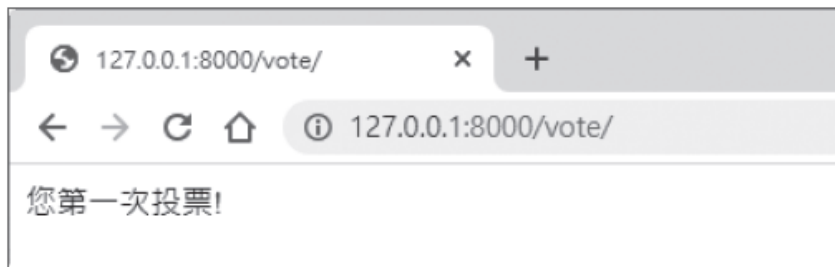
Session 字典

例如：定義一個get_allsession() 自訂程序取得Session 中所有的名稱和內容，存取前最好以「if request.session!=None」檢查Session 是否存在。

```
def get_allsessions(request):  
    if request.session!=None:  
        strsessions=""  
        for key1,value1 in request.session.items():  
            strsessions= strsessions + key1 + ":" + str(value1) + "<br>"  
        return HttpResponse(strsessions)  
    else:  
        return HttpResponse('Session 不存在!')
```

範例：利用Session 防止灌票

剛才我們使用Cookie 技巧寫了一個可以顯示使用者今天瀏覽次數的程式，但是在瀏覽時只要重整一次即執行計次動作，整個瀏覽人次灌了不少水。若是像投票系統，我們希望每一個人在該瀏覽器只能投一票，利用Session 就能防止灌票動作。



6.3.4 Session 的有效時間

參數	意義	預設值
SESSION_EXPIRE_AT_BROWSER_CLOSE	決定 session 是否在瀏覽器關閉時結束。若設為 True，在瀏覽器關閉後，該 session 將會自動結束。	False
SESSION_COOKIE_AGE	session(cookie) 的有效時間。	1,209,600 秒，兩週

Session 預設的有效時間為瀏覽器開啟後宣告Session 啟動開始，持續兩週。可以在<settings.py> 中更改Session 的有效期限。

例如：設定瀏覽器關閉時結束，Session 即自動結束。

```
SESSION_EXPIRE_AT_BROWSER_CLOSE = True
```


例如：設定Session 最大的有效時間為24 分鐘。

```
SESSION_COOKIE_AGE = 1440
```

也可以在<views.py> 函式中，以動態的方式設定Session 的有效時間。語法：

```
request.session.set_expiry(value)
```

參數value 代表持續的時間。

例如：設定Session 持續的時間為5 分鐘。

```
request.session.set_expiry(50*60)
```

6.3.5 刪除 Session

```
del request.session[ 名稱 ]
```

若要刪除目前所有的Session，可以使用clear() 方法：

```
request.session.clear()
```

範例：設定Session 持續時間

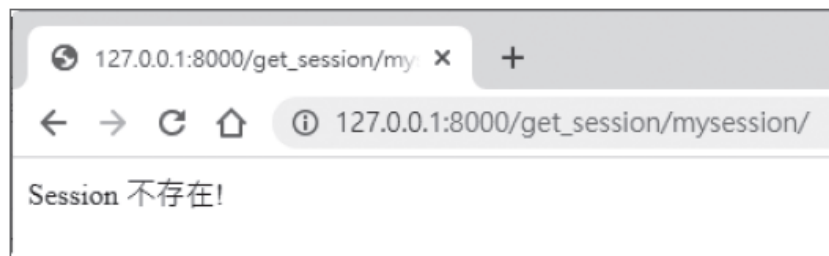
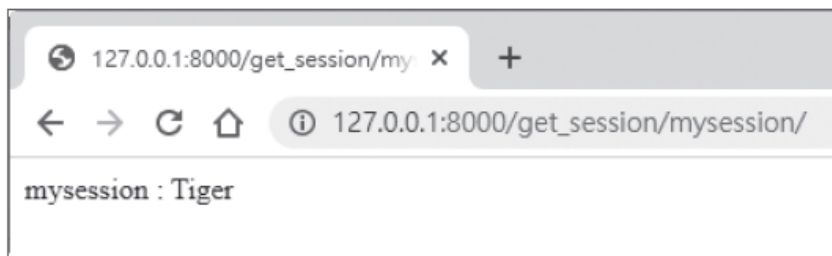
以下範例將示範如何設定Session 持續時間，也可以刪除指定的Session。

輸入「http://127.0.0.1:8000/set_session2/mysession/Tiger/」儲存Session，名稱為「mysession」、內容為「Tiger」，Session 持續時間為30 秒。



關閉瀏覽器後，再重新啟動瀏覽器。輸入

「http://127.0.0.1:8000/get_session/mysession/」，因為Session 持續時間為30 秒，因此在30 秒內可取得名稱為「mysession」的Session 內容，但是逾30 秒就會消失。

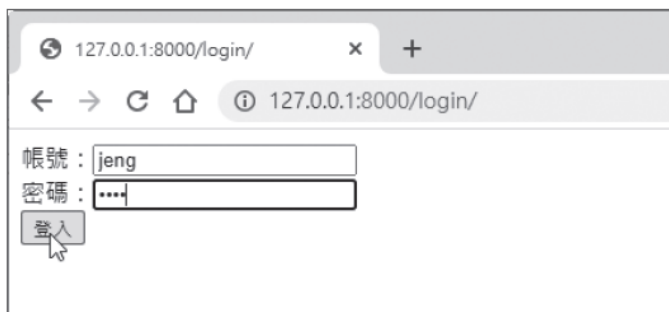


輸入「http://127.0.0.1:8000/delete_session/mysession/」則可以立即刪除名稱為「mysession」的Session 內容。

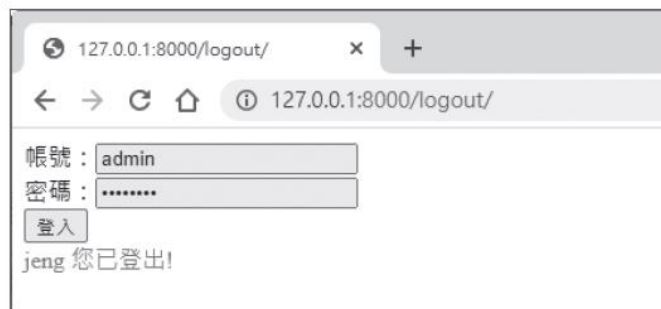
範例：會員系統的登入與登出

以下我們將實作一個會員系統執行登入及登出的動作，會員登入後會產生一個 Session 值來記錄登入者帳號。

輸入「<http://127.0.0.1:8000/login/>」，帳號「jeng」、密碼「1234」後，按登入 鈕即可登入系統，並出現登入成功畫面。



按 登出系統 鈕，可登出系統，並出現登出畫面。



5-3 表單模型化

建立表單模型

在form.py中建立類別 PostForm

```
studentsapp\form.py
```

```
1  from django import forms
2
3  class PostForm(forms.Form):
4      cName = forms.CharField(max_length=20,initial='')
5      cSex = forms.CharField(max_length=2,initial='M')
6      cBirthday = forms.DateField()
7      cEmail = forms.EmailField(max_length=100,initial='',required=False)
8      cPhone = forms.CharField(max_length=50,initial='',required=False)
9      cAddr = forms.CharField(max_length=255,initial='',required=False)
```

建立表單模型物件

```
postform = PostForm()
```

輸出表單

首先建立一個空白物件，並將它傳送給顯示樣版<postform.html>。

```
studentsapp\views.py
```

```
略.....
```

```
def postform(request):
```

```
    略.....
```

```
    postform = PostForm()
```

```
    return render(request, "postform.html", locals())
```

在 <postform.html> 樣版中以 {{postform}} 就可以顯示此空白的 postform 物件。

```
templates\postform.html
```

```
<form action="." method="POST" name="form1">
```

```
    {% csrf_token %}
```

```
    {{postform}}
```

```
</form>
```



Exercise 0: setting.py

常見的Django制式流程

- 設定 `urls.py`

定義網址呼叫的函數，亦稱路由 (route)，指線線、走向。

- 建立函數

建立 `views.py` 檔中所需使用的函數和網址對應；並在函數中決定將資訊送達的網頁。

- Template 模版機制

使用 Django 的 Template 模版機制，建立 `*.html` 檔，供 `views.py` 檔中所需使用的函數傳送資料，並呈現網頁版面。

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```



Exercise 1: set_cookie

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

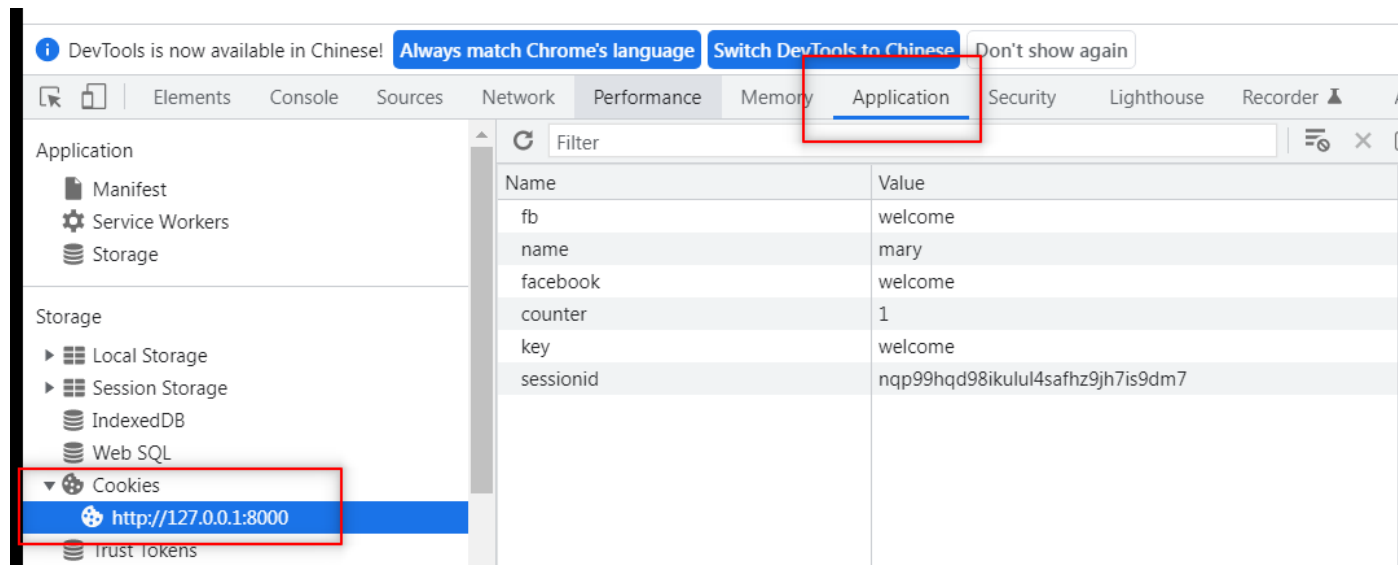
```
- def set_cookie(request, key=None, value=None):  
    response = HttpResponse('Cookie 儲存完畢!')  
    response.set_cookie(key, value)  
    return response
```

Results



Cookie 儲存完畢!

按F12，查看Application





Exercise 2: set_cookie2

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

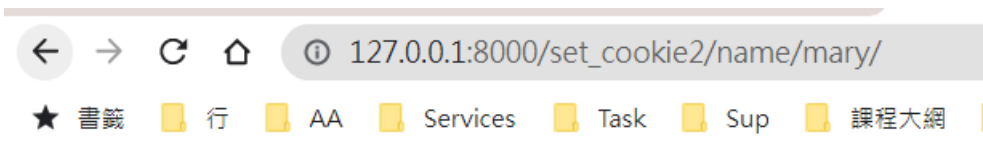
```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

views.py

```
19  def set_cookie2(request, key=None, value=None):  
20      response = HttpResponse('Cookie 有效時間1小時!')  
21      response.set_cookie(key, value, max_age=3600)  
22      return response
```


Results



Cookie 有效時間1小時!

ays match Chrome's language Switch DevTools to Chinese Don't show again

Network Performance Memory Application Security Lighthouse Recorder Adblock Plus						
Filter						
☰ × <input type="checkbox"/> Only show cookies with an issue						
Name	Value	Domain	Path	Expires / Max-Age	S..	
fb	welcome	127.0.0.1	/	Session	9	
name	mary	127.0.0.1	/	2022-03-21T18:31:23.036Z	8	
facebook	welcome	127.0.0.1	/	Session	15	
counter	1	127.0.0.1	/	2022-03-22T00:00:00.537Z	8	
key	welcome	127.0.0.1	/	Session	10	
sessionid	nqp99hqd98ikulul4safhz9jh7is9dm7	127.0.0.1	/	2022-04-04T14:57:56.445Z	41	



Exercise 3: `get_cookie`

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

views.py

```
def get_cookie(request, key=None):  
    if key in request.COOKIES:  
        return HttpResponse('%s : %s' % (key, request.COOKIES[key]))  
    else:  
        return HttpResponse('Cookie 不存在!')
```

Results



127.0.0.1:8000/get_cookie/name/

name : leonardo



Exercise 4: `get_allcookies`

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

views.py

```
def get_all_cookies(request, key=None):  
    print('get_all_cookies() is invoked')  
    if request.COOKIES  $\neq$  None:  
        strcookies = ''  
        for k1, v1 in request.COOKIES.items():  
            strcookies = strcookies + k1 + ':' + v1 + '<br>  
        return HttpResponse('%s' % (strcookies))  
    else:  
        return HttpResponse('Cookie not exist')
```


Results

sessionid:nqp99hqd98ikulul4safhz9jh7is9dm7
key:welcome
fb:welcome
facebook:welcome
counter:2
name:jack

DevTools is now available in Chinese! [Always match Chrome's language](#) [Switch DevTools to Chinese](#) [Don't show again](#)

Elements Console Sources Network Performance Memory **Application** Security Lighthouse Recorder

Application

- Manifest
- Service Workers
- Storage

Storage

- Local Storage
- Session Storage

Name	Value
counter	2
name	jack
facebook	welcome
fb	welcome
key	welcome
sessionid	nqp99hqd98ikulul4safhz9jh7is9dm7



Exercise 5: delete_cookie

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

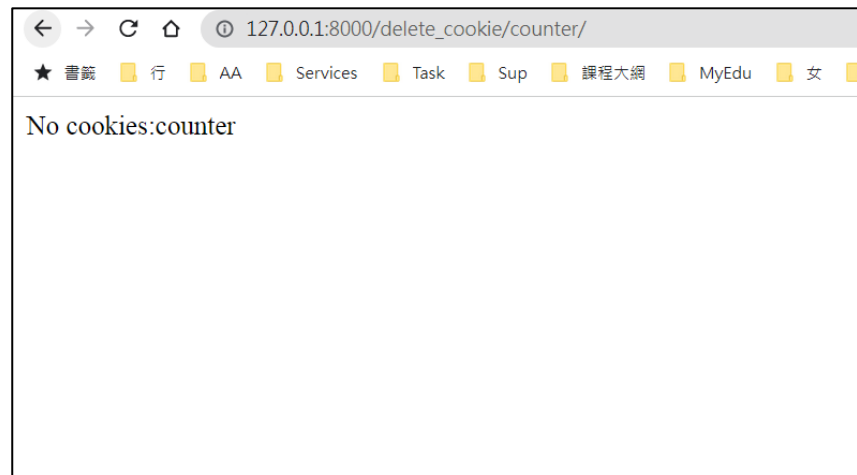
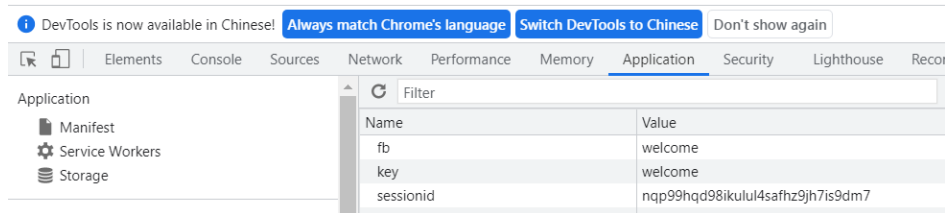
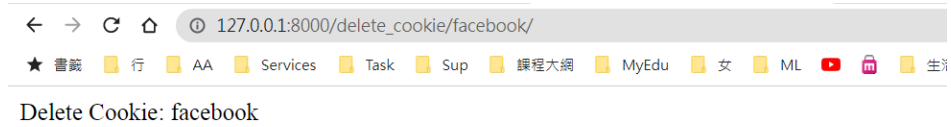
```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

views.py

```
25 def delete_cookie(request, key=None):
26     if key in request.COOKIES:
27         response = HttpResponse('Delete Cookie: ' + key)
28         response.delete_cookie(key)
29         return response
30     else:
31         return HttpResponse('No cookies:' + key)
```

Result





Homework



Exercise 6: index

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

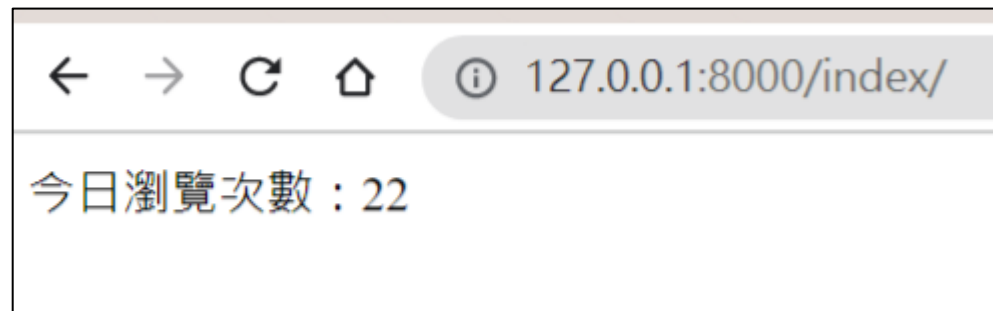
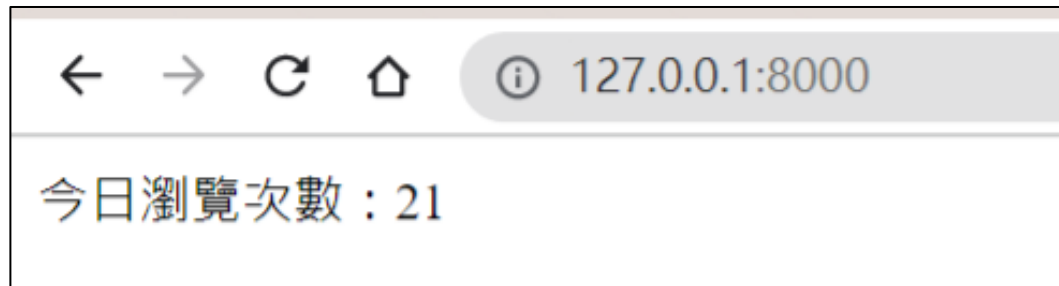
```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```


views.py

```
34 def index(request):
35     if "counter" in request.COOKIES:
36         counter = int(request.COOKIES["counter"])
37         counter += 1
38     else:
39         counter = 1
40     response = HttpResponse('今日瀏覽次數：' + str(counter))
41     tomorrow = datetime.datetime.now() + datetime.timedelta(days=1)
42     tomorrow = datetime.datetime.replace(tomorrow, hour=0, minute=0, second=0)
43     expires = datetime.datetime.strftime(tomorrow, "%a, %d-%b-%Y %H:%M:%S GMT")
44     response.set_cookie("counter", counter, expires=expires)
45     return response
```

Result



Datetime.replace() Function in Python

Note:

- In the replace() we can only pass the parameter the DateTime object is already having, replacing a parameter that is not present in the DateTime object will raise an Error
- It does not replace the original DateTime object but returns a modified DateTime object

Example 1: Replace the current date's year with the year 2000.



Exercise 6: 補充資料

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

Example 1: Replace the current date's year with the year 2000.

Python3

```
# importing the datetime module
import datetime

# Getting current date using today()
# function of the datetime class
todays_date = datetime.date.today()
print("Original Date:", todays_date)

# Replacing the todays_date year with
# 2000 using replace() function
modified_date = todays_date.replace(year=2000)
print("Modified Date:", modified_date)
```

Output:

Original Date: 2021-07-27

Modified Date: 2000-07-27

Directive or format code	Returned Valued	Example
%Y	Full year with century	2021,2022
%y	Year without century with zero padded value	00,01,...,21,22...,99
%-y	Year without century	0,1...,99
%m	Month with zero padded value	01-12
%-m	Month without zero padded value	1-12
%B	Full month name	January, February,..., December
%b	Short form of month	Jan, Feb,...,Dec
%A	Full weekday name	Sunday, Monday,..
%a	Short form of weekday name	Sun, Mon,..
%w	Weekday as decimal value	0-6
%d	Days with zero padded value	01-31
%-d	Days with decimal value	1-31
%H	Hour (24-hour clock) as a zero-padded value.	00-23
%-H	Hour (24-hour clock) without zero-padded value.	0,1,...,23
%I	Hour (12-hour clock) as a zero-padded value.	01-12
%-I	Hour (12-hour clock) without zero-padded value.	1-12
%M	Mins with zero-padded	00-59
%-M	Mins without zero padded value	0-59

Directive or format code	Returned Valued	Example
%S	Secs with zero padded value	00-59
%-S	Secs without zero padded value	0-59
%f	Micro Secs with zero-padded value	000000 – 999999
%p	Locale's AM or PM.	AM/PM
%j	Day of the year with zero padded value	001-366
%-j	Day of the year without zero padded value	1-366
%z	UTC offset in the form +HHMM or -HHMM.	
%Z	Time zone name.	
%C	Locale's appropriate date and time	Fri Apr 02 02:09:07 2020
%x	Locale's appropriate date	02/04/22
%X	Locale's appropriate time	02:04:22
%W	Week number of the year. Monday as first day of week	00-53
%U	Week number of the year. Sunday as first day of week	00-53



Exercise 7: set_sesison

urls.py

```
20 urlpatterns = [  
21     path('admin/', admin.site.urls),  
22     path('set_cookie/<str:key>/<str:value>/', set_cookie),  
23     path('set_cookie/<str:key>/<str:value>/', set_cookie2),  
24     path('get_cookie/<str:key>/', get_cookie),  
25     path('get_allcookies/', get_all_cookies),  
26     path('del_cookie/<str:key>', delete_cookie),  
27     path('', index),  
28     path('index/', index),  
29     path('set_session/<str:key>/<str:value>/', set_session),  
30     path('set_session2/<str:key>/<str:value>/', set_session2),  
31     path('get_session/<str:key>/', get_session),  
32     path('vote/', vote),  
33 ]
```

views.py

```
58 def set_session(request, key=None, value=None):  
59     response = HttpResponse('Session 儲存完畢!')  
60     request.session[key] = value  
61     return response
```

Result



Session 儲存完畢!



Exercise 8: `get_sesison`

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

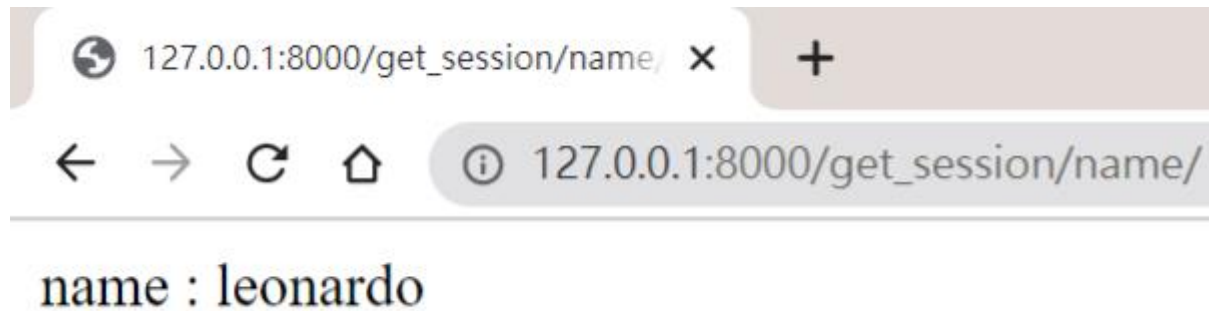
```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

views.py

```
64 def get_session(request, key=None):
65     if key in request.session:
66         return HttpResponse('%s : %s' % (key, request.session[key]))
67     else:
68         return HttpResponse('Session 不存在!')
```

Result





Exercise 9: `get_allseasons`

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

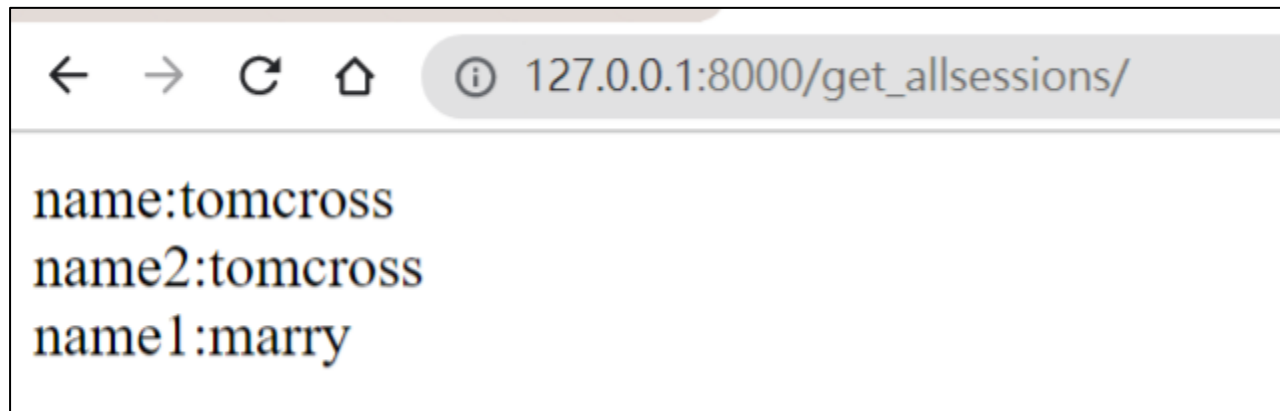
```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

views.py

```
81 def get_allsessions(request):
82     if request.session ≠ None:
83         strsessions = ""
84         for key1, value1 in request.session.items():
85             strsessions = strsessions + key1 + ":" + str(value1) + "<br>"
86         return HttpResponse(strsessions)
87     else:
88         return HttpResponse('Session 不存在!')
```

Result





Exercise 9: vote

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

views.py

```
71 def vote(request):  
72     if not "vote" in request.session:  
73         request.session["vote"] = True  
74         msg = "您第一次投票!"  
75     else:  
76         msg = "您已投過票!"  
77     response = HttpResponseRedirect(msg)  
78     return response
```

Result

← → ↻ 🏠 ⓘ 127.0.0.1:8000/vote/

您第一次投票!

← → ↻ 🏠 ⓘ 127.0.0.1:8000/vote/

您已投過票!

The screenshot shows the Chrome DevTools Application tab. The left sidebar lists storage areas: Local Storage, Session Storage, IndexedDB, Web SQL, Cookies, and Trust Tokens. The 'Cookies' section is highlighted with a red box. The main panel shows a table of storage items. The 'sessionid' item is highlighted with a red box, showing its value as 'ske15wbl4inrufojrjtxav47bbabwuu'.

Name	Value
sessionid	ske15wbl4inrufojrjtxav47bbabwuu



Exercise 10: set_sesison2

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

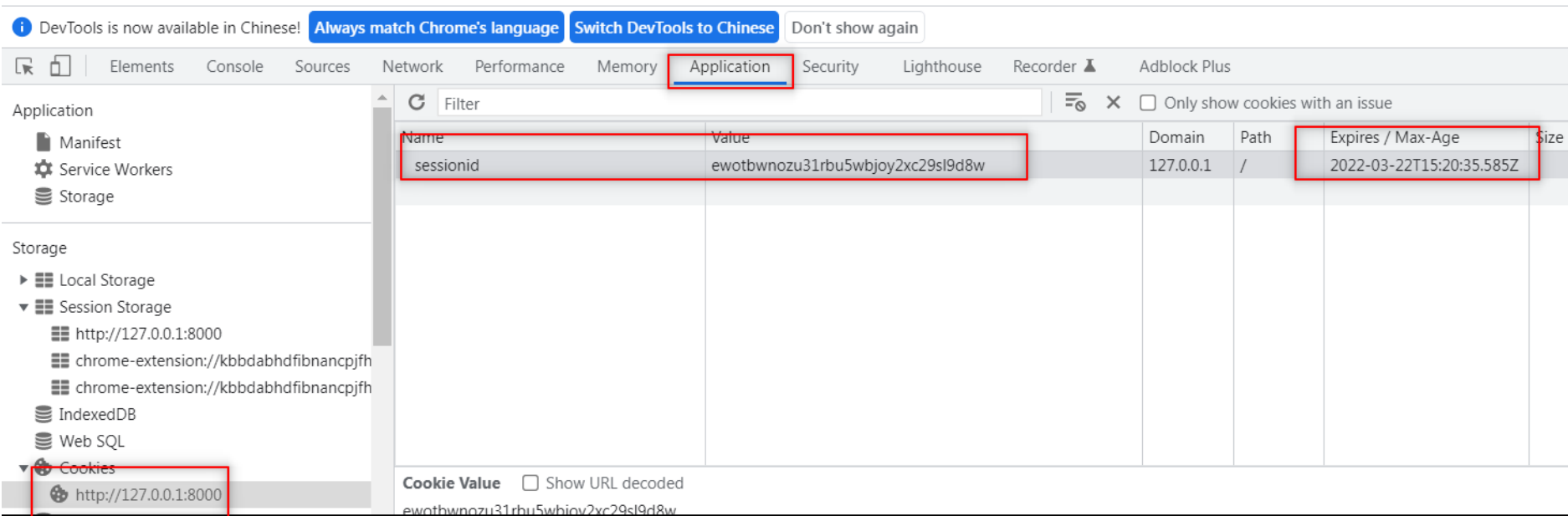
```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

views.py

```
91 def set_session2(request, key=None, value=None):  
92     response = HttpResponse('Session 儲存完畢!')  
93     request.session[key] = value  
94     request.session.set_expiry(30) # 設持續的時間為30秒  
95     return response
```

Result





Exercise 11: login & logout

urls.py

```
16 from django.contrib import admin
17 from django.urls import path
18 from CookieSessionApp import views
```

```
19 |
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('set_cookie/<str:key>/<str:value>/', views.set_cookie),
23     path('set_cookie2/<str:key>/<str:value>/', views.set_cookie2),
24     path('get_cookie/<str:key>/', views.get_cookie),
25     path('get_allcookies/', views.get_allcookies),
26     path('delete_cookie/<str:key>/', views.delete_cookie),
27
28     path('', views.index),
29     path('index/', views.index),
30
31     path('set_session/<str:key>/<str:value>/', views.set_session),
32     path('get_session/<str:key>/', views.get_session),
33     path('get_allsessions/', views.get_allsessions),
34
35     path('vote/', views.vote),
36     path('set_session2/<str:key>/<str:value>/', views.set_session2),
37     path('delete_session/<str:key>/', views.delete_session),
38
39     path('login/', views.login),
40     path('logout/', views.logout),
```

```
41 ]
```

```
from django.contrib import admin
from django.urls import path
from CookieSession.views import *
```

views.py

```
107 def login(request):
108     # 預設帳號密碼
109     username = "jeng"
110     password = "1234"
111     if request.method == 'POST':
112         if not 'username' in request.session:
113             if request.POST['username'] == username and request.POST['password'] == password:
114                 request.session['username'] = username # 儲存Session
115                 message = username + " 您好，登入成功！"
116                 status = "login"
117         else:
118             if 'username' in request.session:
119                 if request.session['username'] == username:
120                     message = request.session['username'] + " 您已登入過了！"
121                     status = "login"
122     return render(request, 'login.html', locals())
```

views.py

```
125 def logout(request):  
126     if 'username' in request.session:  
127         message = request.session['username'] + ' 您已登出!'  
128         del request.session['username'] # 删除Session  
129     return render(request, 'login.html', locals())
```


login.html

```
1 <html>
2 <form action="/login/" method="POST" name="form1">
3     {% csrf_token %}
4     {% if status = "login" %}
5         <div><a href="/logout/">登出系統</a></div>
6     {% else %}
7         <div>帳號 : <input type="text" name="username" id="username" /></div>
8         <div>密碼 : <input type="password" name="password" id="password" /></div>
9         <div><input type="submit" name="button" id="button" value="登入" /></div>
10    {% endif %}
11    <span style="color:red">{{message}}</span>
12 </form>
13 </html>
```