



Cookies

Cookies



- Definición:
 - Texto que un servidor web puede almacenar en el computador del cliente
 - Permite a un sitio web **almacenar** información en el computador del cliente y luego puede recuperarla
 - RFC: <https://www.rfc-editor.org/rfc/rfc6265>
- Se almacenan de forma: nombre-valor
- Se asignan usando el header HTTP:
 - Servidor → cliente web
Set-Cookie: SID=31d4d96e407aad42
 - Cliente web → Servidor
Cookie: SID=31d4d96e407aad42

Cookies



- Asignar usando path y domain:

- Servidor → cliente web

`Set-Cookie: SID=31d4d96e407aad42; Path=/;
Domain=example.com`

- Cliente web → Servidor

`Cookie: SID=31d4d96e407aad42`

Cookies



- Múltiples cookies
 - Servidor → Cliente web

```
Set-Cookie: SID=31d4d96e407aad42; Path=/; Secure; HttpOnly  
Set-Cookie: lang=en-US; Path=/; Domain=example.com
```

- Cliente web → Servidor

```
Cookie: SID=31d4d96e407aad42; lang=en-US
```

Cookies



- Persistentes, usar fecha de expiración en el futuro:
 - Servidor → Cliente web

```
Set-Cookie: lang=en-US; Expires=Wed, 09 Jun 2031 10:18:14 GMT
```

- Cliente web → Servidor

```
Cookie: SID=31d4d96e407aad42; lang=en-US
```

Cookies



- Borrar: asignar fecha en el pasado
 - Servidor → Cliente web
 - `Set-Cookie: lang=; Expires=Mon, 03 Apr 2023 08:49:37 GMT`
 - Cliente web → Servidor

`Cookie: SID=31d4d96e407aad42`

Flask: Cookies



- Para acceder a las cookies que envía el cliente podemos usar el atributo `request.cookies`
 - Es un diccionario con todas las cookies que el cliente envió en la petición HTTP

```
from flask import request
```

```
@app.route('/')
```

```
def index():
```

```
    username = request.cookies.get('username')
```

```
    # use cookies.get(key) instead of cookies[key] to not get a  
    # KeyError if the cookie is missing.
```

Flask: Cookies



- Para asignar cookies que se transmiten al cliente se usa `response.set_cookie`
 - Se usa la función “`make_response()`” para obtener un objeto response

```
from flask import make_response

@app.route('/')
def index():
    resp = make_response(render_template(...))
    resp.set_cookie('username', 'the username')
    return resp
```


Flask: Cookies



- API `set_cookie`

```
set_cookie(key, value='', max_age=None,  
expires=None, path='/', domain=None,  
secure=False, httponly=False,  
samesite=None)
```

- Parámetros

- Key (str): nombre de la cookie
- Value (str): valor
- max_age (opt): número de segundos que debe vivir la cookie en el navegador. Con valor cero será válida por la sesión.
 - Navegadores antiguos (IE 6, 7, 8) pueden no soportar max_age

Flask: Cookies



- Parámetros (continuación):
 - Expires (opt): fecha de expiración de la cookie, debe ser un objeto datetime o timestamp UNIX
 - Esta fecha es relativa al cliente, no al servidor
 - Algunos navegadores pueden “restaurar” que almacena pestañas para un uso futuro
 - También mantienen las cookies de sesión como si nunca se hubiese cerrado el navegador
 - Si se define max_age y expires, max_age tiene mayor importancia
 - Path (opt): limita la cookie para URL relativas a esa ruta. De forma predeterminada es para todo /

Flask: Cookies



- Parámetros (continuación):
 - Domain(opt, str): para definir una cookie válida para subdominios
 - Domain=".example.com" la cookie estará disponible para www.example.com, foo.example.com, etcétera
 - Si no se define, estará disponible solo para el dominio que la asignó
 - Secure (bool): Si es True estará disponible solo para conexiones HTTPS
 - Desde chrome 52 y firefox 52 sitios con HTTP no pueden definir cookies con Secure: True.

Flask: Cookies



- Parámetros (continuación):
 - httponly (bool): Si es True, deshabilita el acceso a la cookie por medio de Javascript
 - Desde código Javascript se puede usar `document.cookie`
 - Esto ayuda a mitigar ataques Cross-site scripting (XSS)
 - samesite (opt): define si puede ser enviada en peticiones cruzadas.
 - Mitiga ataques de falsificación de solicitudes cruzadas (<https://developer.mozilla.org/es/docs/Glossary/CSRF>)
 - Solicitud same-site: www.web.dev carga imagen desde static.web.dev
 - Solicitud cross-site: a.github.io carga imagen de b.github.io
 - github.io es parte de “public suffix list”:
<https://publicsuffix.org/>

Flask: Cookies



- Asigna cookie con username

```
from flask import Flask, request, make_response, render_template

app = Flask(__name__)

@app.route('/', methods = ['GET'])
def Login():
    return render_template('login.html')

@app.route('/details', methods = ['GET', 'POST'])
def login():
    if request.method == 'POST':
        name = request.form['username']
        output = 'Hi, Welcome ' + name + ' '
        resp = make_response(output)
        resp.set_cookie('username', name)
    return resp

app.run(debug=True)
```

Flask: Cookies



- Contador de visitas

```
from flask import Flask, request, make_response

app = Flask(__name__)
app.config['DEBUG'] = True

@app.route('/')
def contador_visitas():
    # Convertir valor str a entero
    # si no viene la cookie, se asume 0
    count = int(request.cookies.get('contador visitas', 0))
    # incrementar contador
    count = count + 1
    output = 'Usted ha visitado esta página ' + str(count) + ' veces'
    resp = make_response(output)
    resp.set_cookie('contador visitas', str(count))
    return resp

@app.route('/get')
def get_contador_visitas():
    count = request.cookies.get('contador visitas', 0)
    resp = make_response(str(count))
    return resp

app.run()
```