# Cookies in Django

## 1. What Are Cookies?

Cookies are small pieces of data stored on the client’s browser, sent by the server. They are included with every future request to the same server.  
  
Analogy: Think of cookies like a 'note' your server gives the browser:  
‘Here’s something I want you to remember for next time.’

## 2. Why Cookies Are Used

Cookies are used for:

1. Session Management - Remembering if a user is logged in and storing session IDs.

2. Personalization - Saving theme preferences, remembering language settings.

3. Tracking - Analytics (Google Analytics, ad tracking).

## 3. How Cookies Work

1. Client sends a request to the server.  
2. Server responds and sets a cookie.  
3. Browser stores the cookie.  
4. On the next request to the same server, browser automatically sends the cookie back.  
5. Server reads it and uses the stored information.

## 4. Cookies in Django

In Django:

- Set Cookie: response.set\_cookie(key, value, ...)

- Get Cookie: request.COOKIES.get(key)

- Delete Cookie: response.delete\_cookie(key)

## 5. Cookie Properties in Django

* key: Name of the cookie. Must be a string. Used to identify the cookie later.
* value: Value of the cookie (must be a string).
* max\_age: Lifetime of the cookie in seconds. If not set, lasts until browser closes.
* expires: Exact date/time when the cookie expires. Overrides max\_age if both set.
* path: URL path for which the cookie is valid. Default '/' (entire site).
* domain: Domain for which the cookie is valid (e.g., '.example.com').
* secure: If True, cookie is sent only over HTTPS.
* httponly: If True, JavaScript cannot access the cookie (prevents XSS).
* samesite: Controls cross-site sending. 'Strict', 'Lax', or 'None' (None requires secure=True).

## 6. Setting a Cookie with All Properties

from django.http import HttpResponse  
import datetime  
  
def set\_cookie\_example(request):  
 response = HttpResponse("Cookie with all properties set!")  
 expires\_time = datetime.datetime.utcnow() + datetime.timedelta(days=1)  
  
 response.set\_cookie(  
 key="username", # Cookie name  
 value="JohnDoe", # Cookie value  
 max\_age=3600, # 1 hour  
 expires=expires\_time, # Expire at this datetime  
 path="/", # Available for entire site  
 domain=None, # Current domain only  
 secure=True, # Only over HTTPS  
 httponly=True, # JS cannot access  
 samesite="Lax" # Partial cross-site protection  
 )  
 return response

## 7. Getting a Cookie in Django

from django.http import HttpResponse  
  
def get\_cookie\_view(request):  
 username = request.COOKIES.get("username")  
 if username:  
 return HttpResponse(f"Hello {username}, welcome back!")  
 else:  
 return HttpResponse("No cookie found.")

## 8. Deleting a Cookie

from django.http import HttpResponse  
  
def delete\_cookie\_view(request):  
 response = HttpResponse("Cookie Deleted!")  
 response.delete\_cookie("username")  
 return response

## 9. Security Notes

- Never store sensitive data like passwords in cookies.

- Use httponly=True to prevent JavaScript access.

- Use secure=True to send cookies only over HTTPS.

- Use samesite='Strict' or 'Lax' to help prevent CSRF.

✅ Summary Table

|  |  |
| --- | --- |
| Action | Django Method |
| Set Cookie | response.set\_cookie() |
| Get Cookie | request.COOKIES.get() |
| Delete Cookie | response.delete\_cookie() |