

1 Session কী?

- Server-side data storage per user.
 - Browser stores only session ID in cookie (except cookie-based backend).
 - Allows storing arbitrary data for anonymous or logged-in users.
-

2 Enabling Sessions

1. MIDDLEWARE:

```
'django.contrib.sessions.middleware.SessionMiddleware'
```

2. INSTALLED_APPS:

```
'django.contrib.sessions'
```

3. Database session হলে migrate করতে হবে:

```
python manage.py migrate
```

3 Session Engines

Engine	Description
db	Database-backed (default)
cache	Cache-only
cached_db	Cache + database
file	File-based sessions
signed_cookies	Data stored in client cookie (signed, not encrypted)

Set in settings.py:

```
SESSION_ENGINE = "django.contrib.sessions.backends.db"
SESSION_COOKIE_AGE = 1209600 # 2 weeks
SESSION_EXPIRE_AT_BROWSER_CLOSE = False
```

4 Session Methods

Dictionary-like methods:

```
__getitem__(key)           # request.session['key']
__setitem__(key, value)    # request.session['key'] = value
__delitem__(key)           # del request.session['key']
__contains__(key)          # 'key' in request.session
get(key, default=None)    # request.session.get('key', default)
update(dict)               # request.session.update({'key': value})
pop(key, default)          # request.session.pop('key', default)
keys()                     # request.session.keys()
values()                   # request.session.values()
items()                    # request.session.items()
has_key(key)               # request.session.has_key('key')
setdefault(key, default)   # request.session.setdefault('key', default)
clear()                    # request.session.clear()
```

Async methods (Django 5.1+):

```
aget(key, default)         # async get
aset(key, value)           # async set
aupdate(dict)              # async update
apop(key, default)         # async pop
akeys()                    # async keys
avalues()                  # async values
aitems()                   # async items
ahas_key(key)              # async has_key
asetdefault(key, default)  # async setdefault
aflush()                   # async flush
aset_test_cookie()          # async set test cookie
atest_cookie_worked()      # async test cookie worked
adelete_test_cookie()       # async delete test cookie
aset_expiry(value)          # async set expiry
aget_expiry_age()           # async get expiry age
aget_expiry_date()          # async get expiry date
aget_expire_at_browser_close() # async check expire at browser close
```

```
a_clear_expired()          # async clear expired sessions  
a_cycle_key()              # async cycle key
```

Other important methods:

```
flush()                   # delete session data + cookie  
set_test_cookie()         # set test cookie  
test_cookie_worked()      # check if browser accepted test cookie  
delete_test_cookie()      # delete test cookie  
set_expiry(value)         # set session expiry  
get_session_cookie_age() # get SESSION_COOKIE_AGE  
get_expiry_age()          # get remaining expiry in seconds  
get_expiry_date()         # get datetime expiry  
get_expire_at_browser_close() # check if expires on browser close  
cycle_key()               # generate new session key while keeping data  
clear_expired()           # remove expired sessions
```