Django Request Lifecycle – Complete Notes

Step 1: User Sends Request

User sends a request via browser by entering a URL or clicking a button. This creates an HTTP request.

Step 2: Web Server Receives Request

Request first goes to a web server like Nginx or Apache, then passed to WSGI/ASGI server (Gunicorn, uWSGI, Daphne) which communicates with Django.

Step 3: Django Enters – settings.py Load

Django loads settings.py, prepares installed apps, and readies middleware for processing the request.

Step 4: Middleware (Request Phase)

1. Request passes through middleware list (top → bottom).
2. Example: SecurityMiddleware, SessionMiddleware, AuthenticationMiddleware.
3. Middleware can modify or block the request.

Step 5: URL Resolver

Django checks urls.py to match the URL to a view. If no match → 404 error.

Step 6: View Function / Class Runs

View handles business logic: fetching data, processing, and returning an HttpResponse.

Step 7: Template Rendering (Optional)

If the view uses a template, Django renders it with context data to generate HTML.

Step 8: Middleware (Response Phase)

1. Response passes through middleware (bottom → top).
2. Middleware can add headers, cookies, or log the response.

Step 9: Response Sent Back

Response is sent from Django → WSGI/ASGI → Web Server → Browser, which renders the page.

Full Flow Diagram

User Browser → Web Server → WSGI/ASGI → Django Middleware (Request) → URL Resolver → View → Template → Django Middleware (Response) → Web Server → Browser Response

One-Line Summary

Django request lifecycle is the sequence of steps that a request passes through middleware, URL routing, view, and template to return a response to the user.