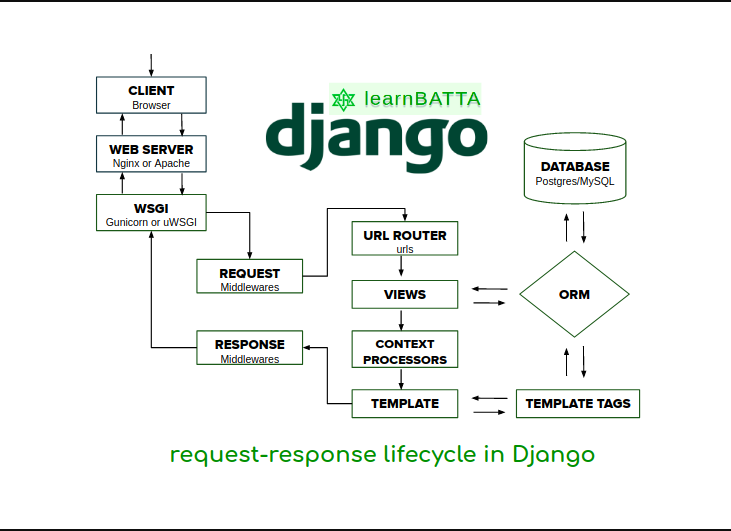
1. Every web application uses HTTP/HTTPS protocol.
2. In HTTP/HTTPS protocol client sends a request to the server and server sends the response back to the client.
3. It's the basic principle of HTTP protocol.
4. While setting up the django application in the server we need a webserver and wsgi server.
5. Webserver helps us in serving static files/content.
6. If we did not use the webserver to serve static files then it has to served by WSGI server which results in more number of requests to the server. So, gradually it slow down the application performance.
7. Web server balances the requests load on the server. So, it's highly recommended to use the webserver.
8. A client can be defined as a piece of software which can send a request by following the HTTP/HTTPS protocol.
9. In general we consider client as a Web Browser.
10. While deploying the django application on the server we use one of the combinations of a "Nginx, uWSGI and Django" or "Nginx, gunicorn and Django" or "Apache, mod\_wsgi and Nginx".
11. As of now, we will discuss the deployment process but we discuss about request-response lifecycle of Django application.
12. We will be discussing on when a request comes in how it can be processed and how the response will be created and sent back to the client.
13. When a client sends a request to the server it first passed to the webserver.
14. Webserver contains the configuration rules to dispatch the request to the WSGI server or served by itself.
15. WSGI server pass the request to the Django application.

Django has the following layers in dealing with request-response lifecycle of a Django application.



Layers of *Django* Application

1. *Request* Middlewares
2. URL Router or URL Dispatcher
3. Views
4. Context Processors
5. Template Renderers
6. *Response* Middlewares