**Middleware & Signals**

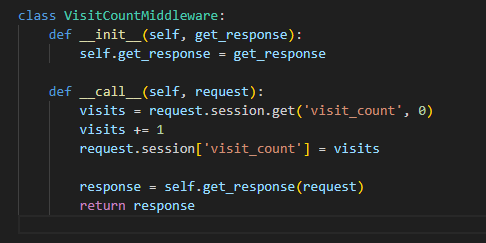
**Middleware:** Think of middleware as a **gatekeeper** or **interceptor** for every HTTP request and response in our app. It’s great when you want to:

* Modify or check **all incoming requests** before they reach your view.
* Modify or add something to **all outgoing responses** before they go to the client.
* Handle things like authentication, logging, session management, or adding headers globally.
* Example use cases:
  + Counting user visits (like the project I gave).
  + Logging request info (IP address, user agent).
  + Enforcing site-wide rules (like blocking banned IPs).
  + Adding security headers to responses.

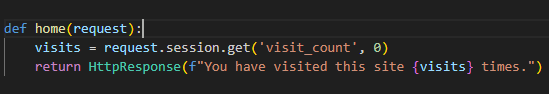
**In short:** Use middleware when you want to affect or observe *every* request/response that passes through your site.

Example. How many times user visit sites count that using middleware.

**Inside app create one file middleware.py**



**View to see logic**



**Add configuration to settings.py**

App.middleware.middlewareclass -> tracker.middleware.VisitCountMiddleware

You are good to go!!

**Signals:**

Signals are like **event listeners** that run some code automatically when something happens in Django. They are useful when you want to:

* React to specific **events** without tightly coupling your code.
* Run some extra code when a model is saved, deleted, or when a request finishes, etc.
* Example use cases:
  + Send a welcome email after a new user registers (listen to the post\_save signal on User).
  + Clear cache after a model changes.
  + Log activity when a request completes (request\_finished signal).
  + Update related models automatically after one changes.

**In short:** Use signals when you want to **react to certain events** anywhere in app, without changing the main flow explicitly.

Example. Send a welcome email after a new user registers.

**This will print emails to the console instead of sending them. Add it to settings.py**

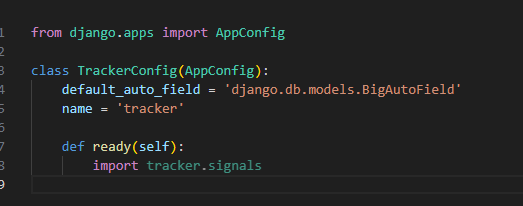


**Init.py -> apps.py -> signals.py**

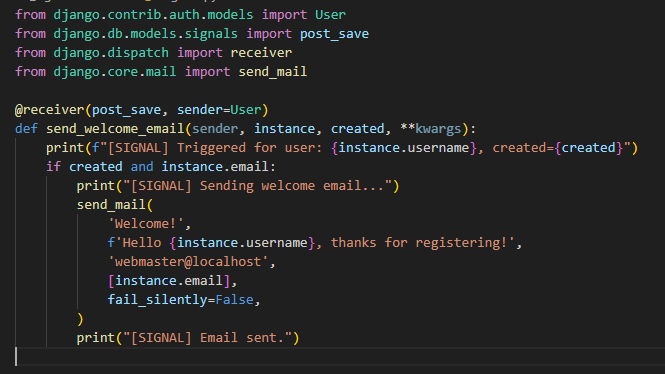
Init.py



apps.py



signals.py



Inside init file call call config which will add to apps.py file and inside apps.py file we will call signals

**Summary:**

* Use **middleware** to **process every request/response globally** before or after views.
* Use **signals** to **react to specific events** (like saving a model or finishing a request) in a decoupled way.