

Django Middlewares - Simple Notes

What is Middleware?

Middleware is a layer between request and response in Django. It processes requests before they reach the view and responses before they go to the browser.

Purpose of Middleware

To perform common tasks like:

- Authentication check
- Logging requests
- Handling exceptions
- Modifying responses
- Security checks (like clickjacking protection)

Structure of Middleware

```
class MyMiddleware:  
    def __init__(self, get_response):  
        self.get_response = get_response # one-time setup  
  
    def __call__(self, request):  
        # Code before view (request processing)  
        response = self.get_response(request)  
        # Code after view (response processing)  
        return response
```

How it works

1. Request → passes through all middlewares → reaches the view.
2. View returns response → passes back through middlewares → client.

Examples

- AuthenticationMiddleware – adds user info to request.
- SessionMiddleware – manages user sessions.
- CsrfViewMiddleware – prevents CSRF attacks.
- ClickjackingMiddleware – adds X-Frame-Options header.

Add Middleware

In settings.py:

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',
```

```
'myapp.middleware.MyMiddleware',
]
```

⚡ Summary

Step	Purpose
<code>_init_ / _call_</code>	Used to process request and response globally