

System Requirements

Specification Index

For

Implementing a Middleware to Restrict Access Based on User IP Addresses

(Topic:- Django Middleware)

Version 1.0

Problem Statement

Scenario:

You have joined a tech company as a Django developer, and your task is to implement a middleware that restricts access to the website based on user IP addresses. The company wants to ensure that only users from certain regions or IPs can access the website, while all others should receive a 403 Forbidden response.

The middleware should:

- Allow access to a predefined list of allowed IP addresses.
- Block requests coming from IPs that are not in the allowed list.
- Log the attempts of unauthorized access with details like the IP address and the reason for blocking.
- Handle edge cases like missing IP addresses or invalid formats.

Your task:

Implement a middleware class `RestrictAccessMiddleware` that performs the following checks:

- If the user's IP address is in the allowed list, they can access the website.
- If the user's IP address is not allowed, return a 403 response.
- Ensure that the middleware handles missing or invalid IPs gracefully.

Add the middleware to the Django project settings.

Execution Steps to Follow:

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to

Application menu(Three horizontal lines at left top)->Terminal->NewTerminal.

3. The editor Auto Saves the code.
4. If you want to exit (logout) and to continue the coding later anytime(using Save & Exit option on Assessment LandingPage) then you need to use CTRL+Shift+B command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while

logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.

6. To test any Restful application, the last option on the left panel of IDE, you can find

ThunderClient, which is the lightweight equivalent of POSTMAN.

7. To test any UI based application the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

8. Install 'djangoestframework' module before running the code. For this use the following command.
`pip install djangoestframework`
9. Use the following command to run the server
`python3 manage.py runserver`
10. Mandatory: Before final submission run the following commands to execute testcases
`python3 manage.py test library.test.test_functional`
`python3 manage.py test library.test.test_exceptional`
`python3 manage.py test library.test.test_boundary`
11. To test rest end points
Click on 'Thunder Client' or use Ctrl+Shift+R->Click on 'New Request' (at left side of IDE)
12. Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.
13. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.