

Day 48



Shutil Module in Python:

What Is shutil?

shutil stands for shell utilities. It provides high-level file and directory operations, such as:

- Copying files
- Moving files
- Deleting directories
- Archiving (zip, tar)
- Checking disk usage

It works on top of lower-level modules like os and pathlib.

Importing shutil

```
import shutil
```

Copying Files: shutil.copy() Copies file contents and permissions.

```
1 import shutil
2 shutil.copy("Aditya.txt", "Utsav.txt")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS E:\Python\Day41-50\Day48> python main.py
● PS E:\Python\Day41-50\Day48> cat .\Utsav.txt
Aditya Is a boy.

shutil.copy2() : Copies file with metadata (timestamps).

shutil.copyfile() : Copies only file contents (no metadata).

Moving Files and Directories: shutil.move()

Deleting Directories: shutil.rmtree() : Deletes a directory and everything inside it.

Requests Module in Python:

What Is requests?

requests is a popular Python library used to make HTTP requests easily.

With it, you can:

- Fetch data from APIs
- Send data to servers
- Download files
- Work with JSON responses
- Authenticate requests

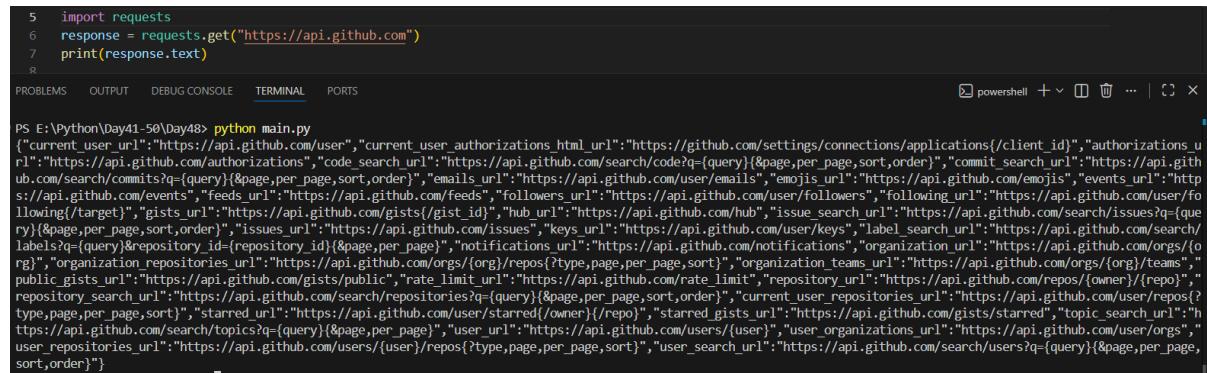
It replaces complex built-in modules like urllib.

Installing requests

requests is not included by default.

Pip install requests

Example: basic GET request.



```
5 import requests
6 response = requests.get("https://api.github.com")
7 print(response.text)
```

The screenshot shows a terminal window titled "powershell". It contains the code above, followed by the output of the command "python main.py". The output is a large JSON object representing the GitHub API response, which includes various URLs for user, repositories, and organization endpoints.

Key attributes:

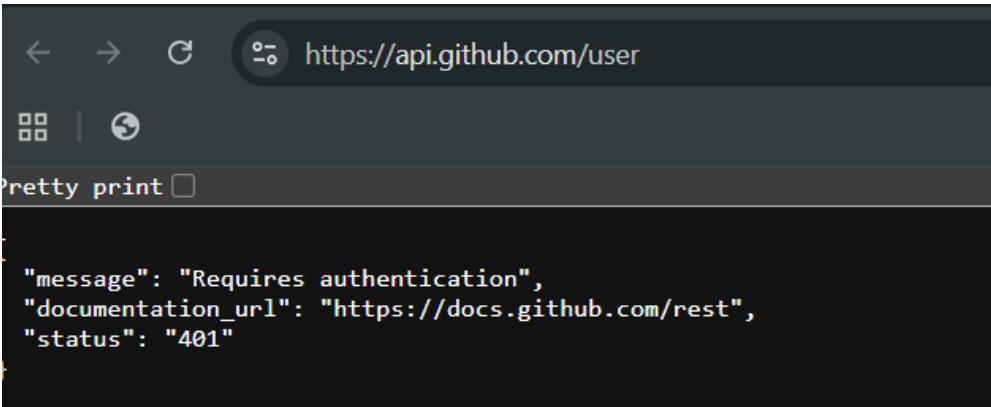
Attribute	Meaning
response.status_code	HTTP status
response.text	Response body (string)
response.json()	JSON → Python dict
response.headers	Response headers
response.url	Final URL

Example: working with JSON data.

```
9 import requests
10 response = requests.get("https://api.github.com")
11 data = response.json()
12 print(data["current_user_url"])
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS E:\Python\Day41-50\Day48> python main.py
https://api.github.com/user



```
{"message": "Requires authentication", "documentation_url": "https://docs.github.com/rest", "status": "401"}
```

Example: making the POST requests.

```
13 import requests
14 data = {
15     "username": "admin",
16     "password": "1234"
17 }
18 response = requests.post("https://httpbin.org/post", data=data)
19 print(response.json())
20
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS E:\Python\Day41-50\Day48> python main.py
{'args': {}, 'data': '', 'files': {}, 'form': {'password': '1234', 'username': 'admin'}, 'headers': {'Accept': '*/*', 'Accept-Encoding': 'gzip, deflate', 'Content-Length': '28', 'Content-Type': 'application/x-www-form-urlencoded'}, 'Host': 'httpbin.org', 'User-Agent': 'python-requests/2.32.5', 'X-Amzn-Trace-Id': 'Root=1-6944dc6b-4b7317ed6b40c7a3ef52793'}, 'json': None, 'origin': '110.224.106.112', 'url': 'https://httpbin.org/post'}

Summary:

- requests makes HTTP easy
- Clean syntax & readable code
- Industry standard for API work
- Essential Python skill

--The End--