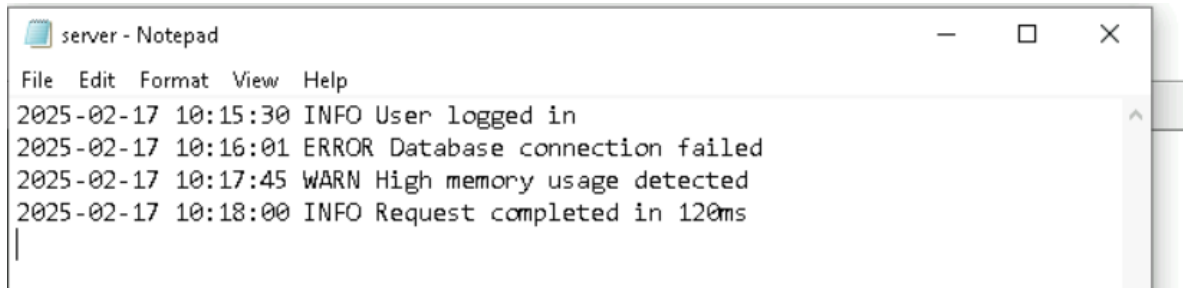


Groovy Projects

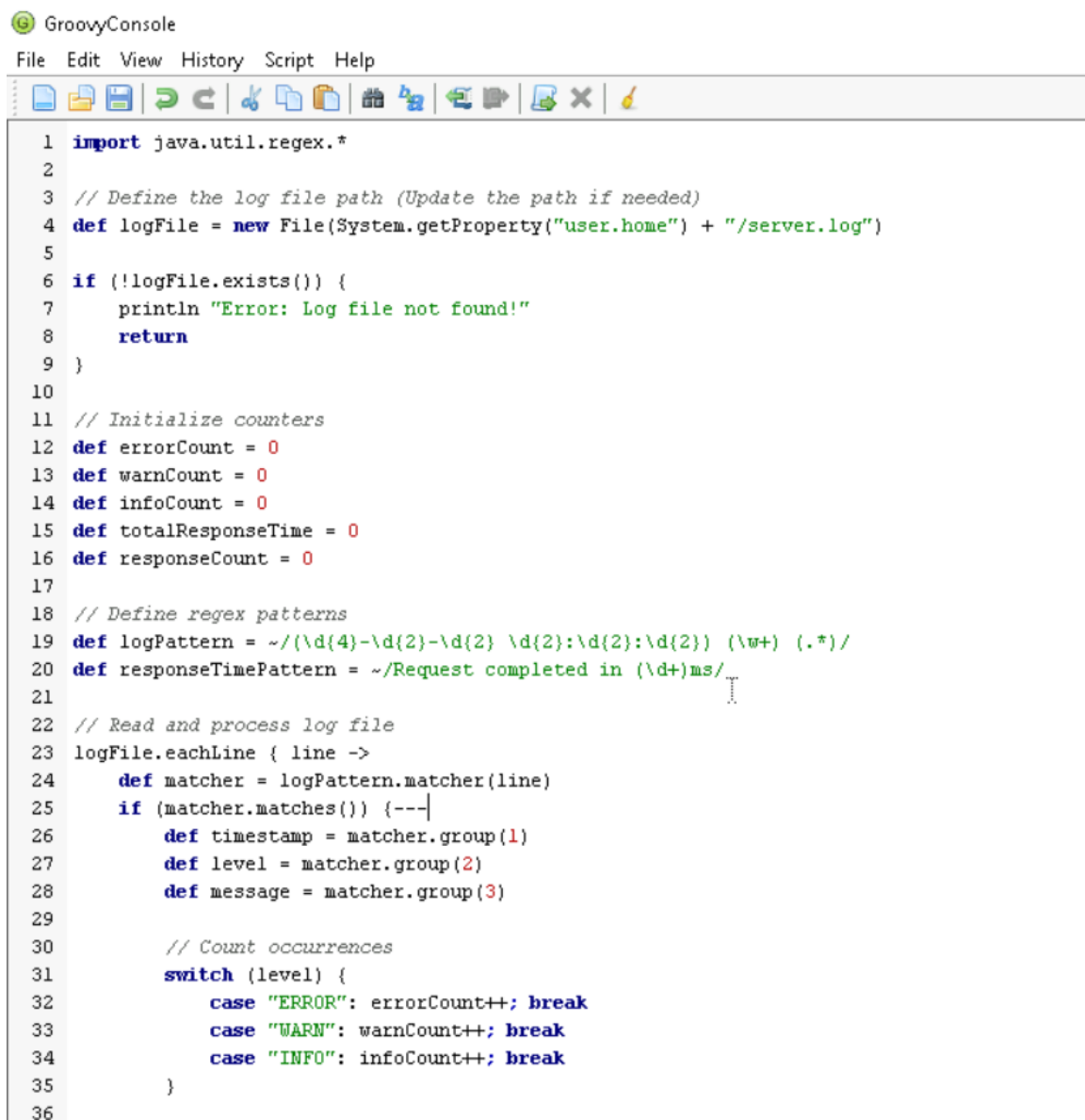
1. Groovy Script for Log File Analysis

1. Open Notepad and paste this info and save in the c://user/Administrator as server.log text file.



```
server - Notepad
File Edit Format View Help
2025-02-17 10:15:30 INFO User logged in
2025-02-17 10:16:01 ERROR Database connection failed
2025-02-17 10:17:45 WARN High memory usage detected
2025-02-17 10:18:00 INFO Request completed in 120ms
```

2. Right the script in the Groovy Console



```
GroovyConsole
File Edit View History Script Help

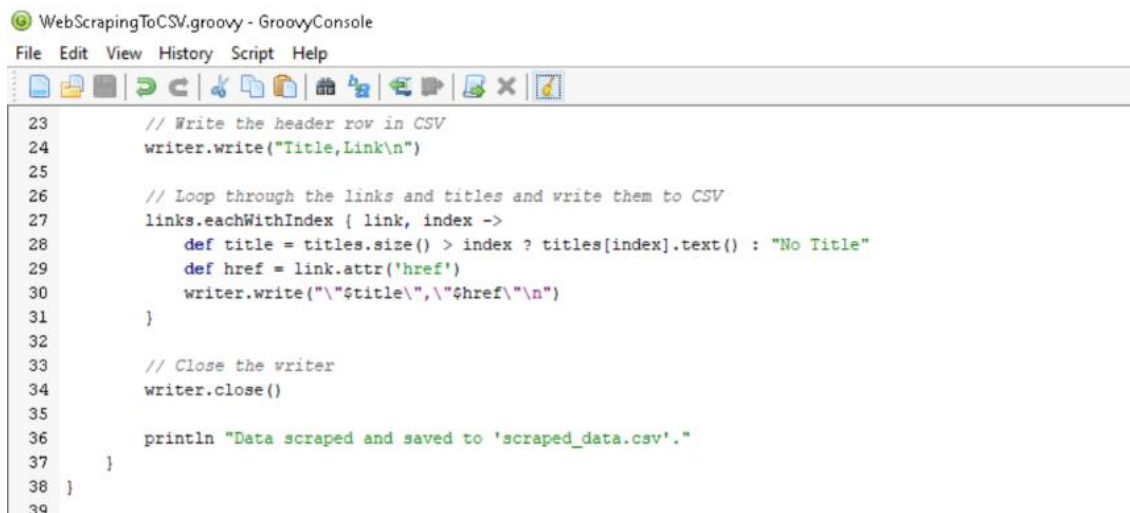
1 import java.util.regex.*
2
3 // Define the log file path (Update the path if needed)
4 def logFile = new File(System.getProperty("user.home") + "/server.log")
5
6 if (!logFile.exists()) {
7     println "Error: Log file not found!"
8     return
9 }
10
11 // Initialize counters
12 def errorCount = 0
13 def warnCount = 0
14 def infoCount = 0
15 def totalResponseTime = 0
16 def responseCount = 0
17
18 // Define regex patterns
19 def logPattern = ~/(\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}) (\w+) (.*)/
20 def responseTimePattern = ~/Request completed in (\d+)ms/
21
22 // Read and process log file
23 logFile.eachLine { line ->
24     def matcher = logPattern.matcher(line)
25     if (matcher.matches()) {---|
26         def timestamp = matcher.group(1)
27         def level = matcher.group(2)
28         def message = matcher.group(3)
29
30         // Count occurrences
31         switch (level) {
32             case "ERROR": errorCount++; break
33             case "WARN": warnCount++; break
34             case "INFO": infoCount++; break
35         }
36     }
```

3. Output

```
==== Log Analysis Report ====  
Total INFO logs   : 2  
Total WARN logs   : 1  
Total ERROR logs  : 1  
Average Response Time: 120ms
```

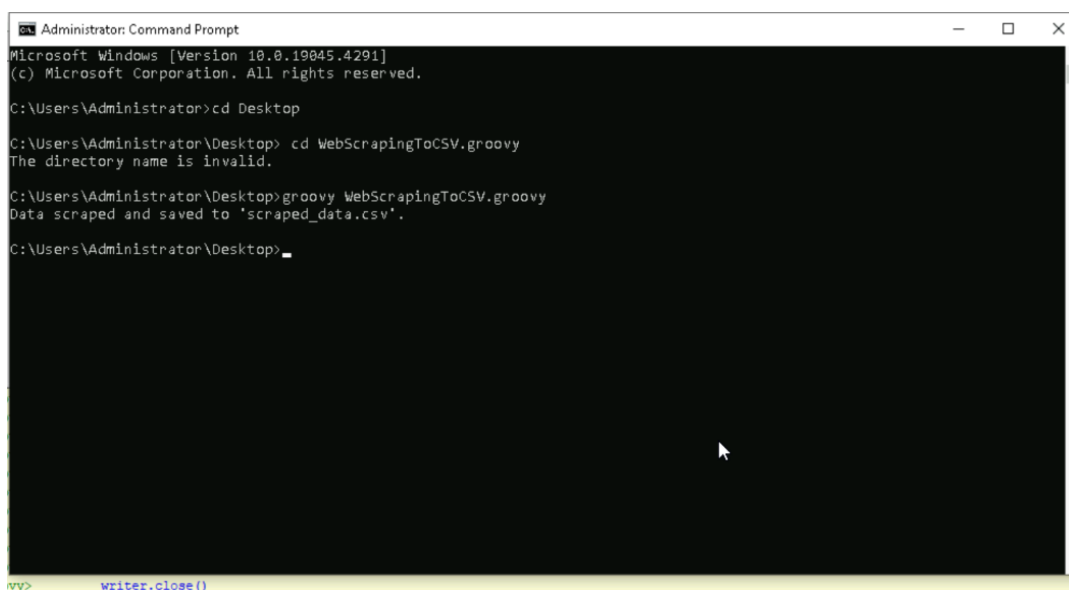
2. Web Scraping with Groovy & Jsoup

Step 1: Run the script in groovy



```
WebScrapingToCSV.groovy - GroovyConsole  
File Edit View History Script Help  
23 // Write the header row in CSV  
24 writer.write("Title,Link\n")  
25  
26 // Loop through the links and titles and write them to CSV  
27 links.eachWithIndex { link, index ->  
28     def title = titles.size() > index ? titles[index].text() : "No Title"  
29     def href = link.attr('href')  
30     writer.write("\"$title\", \"$href\"\n")  
31 }  
32  
33 // Close the writer  
34 writer.close()  
35  
36 println "Data scraped and saved to 'scraped_data.csv'. "  
37 }  
38 }  
39 }
```

Step 2: Run the script in terminal



```
Administrator: Command Prompt  
Microsoft Windows [Version 10.0.19045.4291]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\Administrator>cd Desktop  
C:\Users\Administrator\Desktop> cd WebScrapingToCSV.groovy  
The directory name is invalid.  
C:\Users\Administrator\Desktop>groovy WebScrapingToCSV.groovy  
Data scraped and saved to 'scraped_data.csv'.  
C:\Users\Administrator\Desktop>  
wy> writer.close()
```

Step 3: Output

 scraped_data - Notepad

File Edit Format View Help

Title, Link

"No Title", "https://www.iana.org/domains/example"