Apache logging documentation

1. Access log

Log messages can be found at:

```
/var/log/apache2/access.log
```

after the user have installed and ran the apache webserver.

2. Error log

The error log can be found at:

```
/var/log/apache2/error.log
```

However, the format for the error log is relatively free-form. Making it very hard to extract information.

3. Access log entry:

An example of an access log entry is:

```
127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET
/apache_pb.gif HTTP/1.0" 200 2326 "http://www.example.com/start.html"
"Mozilla/4.08 [en] (Win98; I ;Nav)"
```

Each part of this log entry is described as below:

- `127.0.0.1`: This is the IP address of the client (remote host) which made the request to the server.
- `-`: The "hyphen" in the output indicates that the requested piece of information is not available. In this case, the information that is not available is the RFC 1413 identity of the client determined by `identd` on the clients' machine.
- `frank`: This is the `userid` of the person requesting the document as determined by HTTP authentication
- `[10/Oct/2000:13:55:36 -0700]`: The time that the server finished processing the request. The format is
  `[day/month/year:hour:minute:second zone]`
- `"GET /apache_pb.gif HTTP/1.0"`: The request line from the client is given in double quotes. First, the method used by the client is GET. Second, the client requested the resource /apache_pb.gif, and third, the client used the protocol HTTP/1.0
- **200: This is the status code that the server sends back to the client.**
- `2326`: The last entry indicates the size of the object returned to the client, not including the response headers. If no content was returned to the client, this value will be "-".
- "`http://www.example.com/start.html`": The "Referer" (sic) HTTP request header. This gives the site that the client reports having been referred from.
- `"Mozilla/4.08 [en] (Win98; I ;Nav)"`: The User-Agent HTTP request header. This is the identifying information that the client browser reports about itself.

4. HTTP status code

Overall range	Defined range	Category
100-199	100-101	Informational
200-299	200-206	Successful
300-399	300-305	Redirection
400-499	400-417	Client error
500-599	500-505	Server error

Table 8-5. HTTP status codes overview

Status code	Reason
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request URI Too Long
415	Unsupported Media Type
416	Request Range Not Satisfiable
417	Expectation Failed

Table 8-6. HTTP client error status codes

5. Extracting information from the access log file:

The method shown below returns an ArrayList of httpLogEvents which were extracted from the access log file. The log was read line by line. For each line, anything between quotation marks were trimmed of space characters, the quotation marks were removed. The line will then be split with the space character as the separator into an ArrayList of Strings named lineComponents. The ArrayList will later be used to create the httpLogEvent.

```
public ArrayList<httpLogEvent> getEventsFromApacheHTTPDLogs() throws
IOException {
    BufferedReader reader = new BufferedReader(new
FileReader("/var/log/apache2/access.log"));
    String line = null;
    ArrayList<httpLogEvent> result = new ArrayList<>();

    while ((line = reader.readLine()) != null) {
        Pattern p = Pattern.compile("\"([^\"]*)\"");

        Matcher m = p.matcher(line);
        while (m.find()) {
            line = line.replace(m.group(1), m.group(1).replace(" ", ""));
        }

        ArrayList<String> lineComponents = new
ArrayList<String>(Arrays.asList(line.split(" ")));
        result.add(new httpLogEvent(lineComponents));
    }
    return result;
}
```

6. Event class:

The information from the log file is ordered. Accordingly, it is also extracted in an orderly fashion.

```
public class httpLogEvent {
    String IPAddress;
    String identd;
    String userID;
    String time;
    String timeZone;
    String protocol;
    String statusCode;
    String returnObjSize;
    String referer;
    String clientBrowser;

    public httpLogEvent(ArrayList<String> splittedLog) {
        IPAddress = splittedLog.get(0);
        identd = splittedLog.get(1);
        userID = splittedLog.get(2);
        time = splittedLog.get(3);
        timeZone = splittedLog.get(4);
        protocol = splittedLog.get(5);
        statusCode = splittedLog.get(6);
        returnObjSize = splittedLog.get(7);
        referer = splittedLog.get(8);
```

```
        clientBrowser = splittedLog.get(9);
    }
```

7. References

1. https://www.feistyduck.com/library/apache-security/online/apachesc-CHP-8.html?fbclid=IwAR0w-QyzErVzbyQJuQiVbtIchbGhdcD4ZS9L_sFbHsFblCMGA91MDRWNBLM
2. https://httpd.apache.org/docs/1.3/logs.html