

## 50. Use stored cookies to login information

```
public static void main(String[] args){
    try{
        File file = new File("Cookies.data");
        FileReader fileReader = new FileReader(file);
        BufferedReader Buffreader = new BufferedReader(fileReader);

        String strline;
        while((strline=Buffreader.readLine())!=null){
            StringTokenizer token = new StringTokenizer(strline,";");
            while(token.hasMoreTokens()){
                String name = token.nextToken();
                String value = token.nextToken();
                String domain = token.nextToken();
                String path = token.nextToken();
                Date expiry = null;

                String val;
                if(!(val=token.nextToken()).equals("null")) {
                    expiry = new Date(val);
                }
                Boolean isSecure = new Boolean(token.nextToken()).booleanValue();
                Cookie ck = new Cookie(name,value,domain,path,expiry,isSecure);
                System.out.println(ck);
                driver.manage().addCookie(ck); // This will add the stored cookie to current session
            }
        }
    }catch(Exception ex){
        ex.printStackTrace();
    }
    driver.get("http://demo.guru99.com/test/cookie/selenium_aut.php");
}
```

OUTPUT: You are taken directly to the login success screen without entering the input user id and password

NOTE: Use hard refresh in case you see the login page after executing the above script.

## 51. Do you use JavaScriptExecutor?

- This helps me write my own JavaScript. JS has way more control than selenium.
- we can send JS commands to the browser with using this class  
`JavaScriptExecutor jsExecutor=(JavaScriptExecutor)driver;`
  - `executeScript();` performs the command
  - Inside the parameter is where you put JS code
- `jsExecutor.executeScript("alert('WARNING: This is a useless message');")` → This code will bring up a JS popup
- You can also put 2 parameters is `.executeScript("js code",element);`
  - Used for scrolling (selenium is not good with scrolling, you can say a challenge is when I was working on terms and condition page, where you have to read the page before clicking on continue.
  - When I tried using selenium and actions class it didn't work, so i used `javaexecutor` ) and clicking an element;