Warm Up Program

Write a Java program to download Sur La Table’s recipes and put them into an excel doc (use .csv since its plaintext 😊). (This means “web crawling” starting from <https://www.surlatable.com/recipes/?srule=best-matches&start=0&sz=24> ). You may assume there will only be 96 recipes (instead of 2000+). Advance with start=0, start=24, start=48, etc.

The excel file should columns for: Author, the “path” of the dessert (example is Home/ Recipes/Desserts/Cakes & Custards/Dark Chocolate and Cherry Clafoutis), Name, servings, ingredients, and instructions. Be sure to remove all the html characters from the output.

You must write a program to parse through the HTML source to find the recipe links and then open and parse through each of those recipes to grab the specific pieces of information you are looking for. Use “tall tells” to find what you are looking for. This is not as hard as it sounds one you find those “tall tells”.

This might be a helpful starting point (code is adopted from <https://zetcode.com/java/readwebpage/> ):  
  
import java.io.IOException;  
import java.net.URI;  
import java.net.http.HttpClient;  
import java.net.http.HttpRequest;  
import java.net.http.HttpResponse;  
import java.io.\*;  
  
public class ReadWebPage {  
  
 public static void main(String[] args) throws IOException, InterruptedException {  
  
 HttpClient client = HttpClient.newHttpClient();  
 HttpRequest request = HttpRequest.newBuilder()  
 .uri(URI.create("https://www.surlatable.com/recipes/?srule=best-matches&start=0&sz=24")) //starting URL…  
 .GET() // GET is default  
 .build();  
  
 HttpResponse<String> response = client.send(request,  
 HttpResponse.BodyHandlers.ofString());

//output to a file so its easy to mess with (you won’t be for your finished program)  
 FileOutputStream fs = new FileOutputStream("output.txt");  
 PrintWriter pw = new PrintWriter(fs);  
 pw.println(response.body()); //response.body() is the html source code in a string format. It outputs to a file so you can see it easier right now, but you will ultimately want to just manipulate the strings a lot  
 pw.close();  
 }  
}

**Don’t DOS them: Only issue a request every 10-15 seconds (look up how to sleep your program). There are ~100 requests you may have to issue; your program will take about 15 minutes to run once it is complete (assuming 10 second delay between requests). Don’t assume your program will work correctly the first time. Wow, that means you cannot wait until the last minute since it takes 15 minutes to run!!!**

You program should run on my computer using JGrasp with no changes. Deductions may apply if this is not done correctly.