

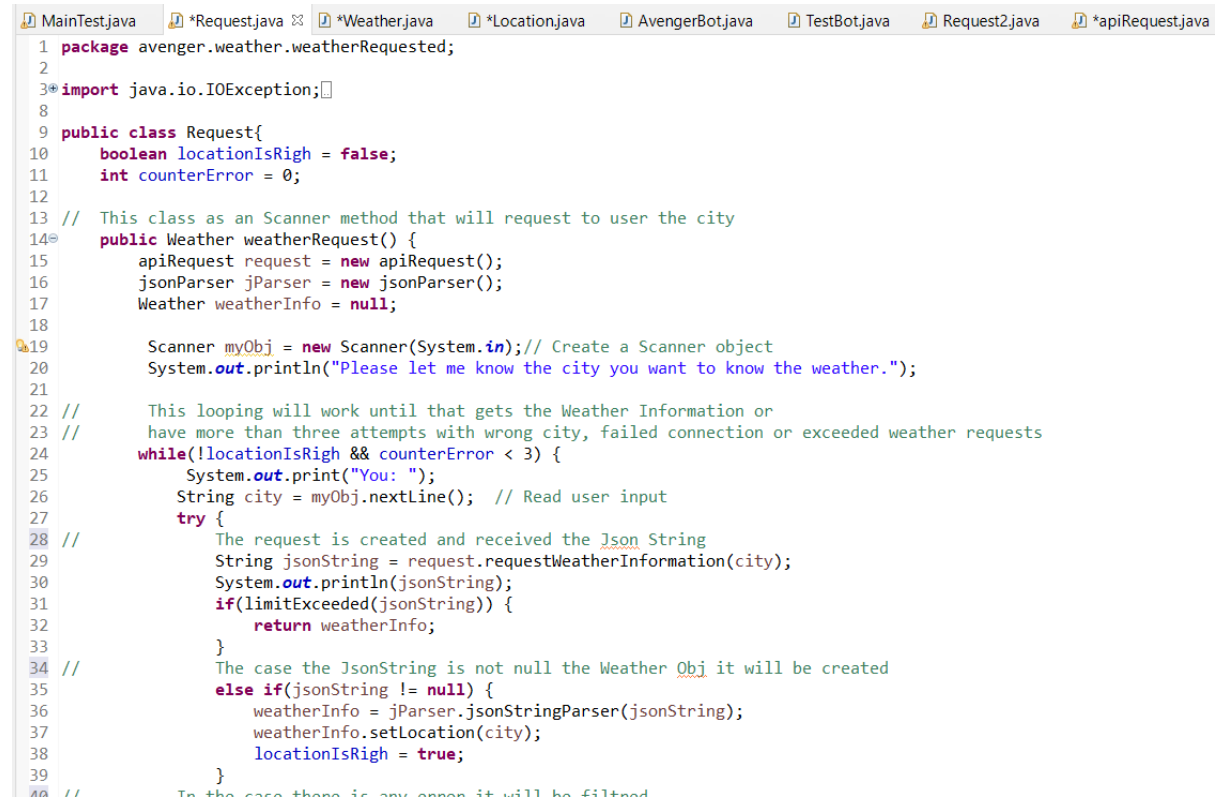
Version 2.0

This version receives a new class that allow to combine the first classes. The Class Request it will use the Parser and API Request Class.

This class it will ask for a City and use to call the APIrequest, check for error, parser the JsonString and then create a Weather Object that will be returned and can be manipulated.

Using this class, we can create a test to show the attributes of the Weather Object.

Screenshot 1.0 – Request



```
1 package avenger.weather.weatherRequested;
2
3 import java.io.IOException;
4
5 public class Request{
6     boolean locationIsRigh = false;
7     int counterError = 0;
8
9     // This class as an Scanner method that will request to user the city
10    public Weather weatherRequest() {
11        apiRequest request = new apiRequest();
12        jsonParser jParser = new jsonParser();
13        Weather weatherInfo = null;
14
15        Scanner myObj = new Scanner(System.in); // Create a Scanner object
16        System.out.println("Please let me know the city you want to know the weather.");
17
18        // This looping will work until that gets the Weather Information or
19        // have more than three attempts with wrong city, failed connection or exceeded weather requests
20        while(!locationIsRigh && counterError < 3) {
21            System.out.print("You: ");
22            String city = myObj.nextLine(); // Read user input
23            try {
24                // The request is created and received the Json String
25                String jsonString = request.requestWeatherInformation(city);
26                System.out.println(jsonString);
27                if(limitExceeded(jsonString)) {
28                    return weatherInfo;
29                }
30                // The case the JsonString is not null the Weather Obj it will be created
31                else if(jsonString != null) {
32                    weatherInfo = jParser.jsonStringParser(jsonString);
33                    weatherInfo.setLocation(city);
34                    locationIsRigh = true;
35                }
36            }
37            // To the case there is any error it will be filtered
38        }
39    }
40}
```

Screenshot 2.0 – Request

```

38         locationIsRight = true,
39     }
40 // In the case there is any error it will be filtered
41 }catch(IOException an) {
42     cityNotFound(counterError);
43 }catch(InterruptedException an) {
44     connectionError(counterError);
45 }catch(NullPointerException e) {
46     cityNotFound(counterError);
47 }
48 }
49 return weatherInfo;
50 }
51
52 // Check if the limit of requests to the API have been achieved
53 private boolean limitExceeded(String jsonString) {
54     String string = ("{"message\":\"You have exceeded the rate limit per minute for your plan, BASIC, \"
55         + "by the API provider\"}");
56     return jsonString.equals(string);
57 }
58 // Check if the city informed was not found
59 private void cityNotFound(int counter){
60     counterError++;
61     if(counterError == 3)
62         System.out.println("Sorry this city is not your database. \n"
63             + "You can try again later.");
64     else {
65         System.out.println("Sorry we can't find this city." );
66         System.out.println("Could you please the spelling and write the name again" );
67     }
68 }
69 // Check if happened a error connection
70 private void connectionError(int counter){
71     counterError++;
72     if(counterError == 3)
73         System.out.println("Sorry we are having a little trouble with the connection \n"
74             + "Please try again later.");
75     else {
76         System.out.println("Sorry we are having a little trouble with the connection");
77         System.out.println("Please try again");
78     }
79 }
80 }
81 }

```

Screenshot – Test

```

1 package avenger.weather.test;
2
3 import java.io.IOException;
4
5 public class MainTest {
6
7     public static void main(String[] args) throws IOException, InterruptedException{
8         Weather weatherInfo = null;
9         Request req = new Request();
10
11         weatherInfo = req.weatherRequest();
12
13         if(weatherInfo != null) {
14             System.out.println("Location: " + weatherInfo.getLocation().getCity() + ", " + weatherInfo.getLocation().getCountry());
15             System.out.println("Current Temperature: " + weatherInfo.getTemp());
16             System.out.println("Feels Like: " + weatherInfo.getFeelsLike());
17             System.out.println("Max: " + weatherInfo.getMax());
18             System.out.println("Min: " + weatherInfo.getMin());
19             System.out.println("Wind Speed: " + weatherInfo.getWindSpeed());
20         } else {
21             System.out.println("Sorry we are unable to locate help you right now!");
22         }
23     }
24 }

```

Console

```

<terminated> MainTest [Java Application] C:\Program Files\Java\jdk-11.0.9\bin\javaw.exe (4 Apr 2021, 14:21:40 – 14:21:45)
Please let me know the city you want to know the weather.
You: Cork
Location: Cork, Ireland
Current Temperature: 13.94
Feels Like: 12.9
Max: 15.0
Min: 13.0
Wind Speed: 4.63

```