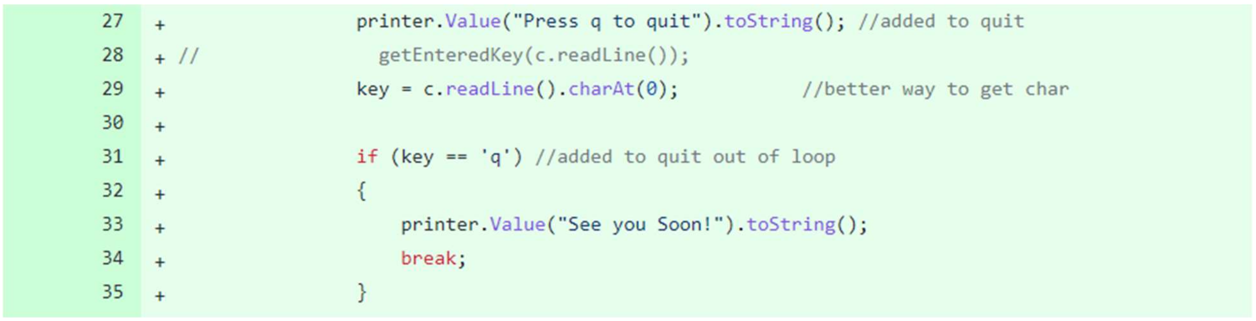
Code Review and Changes Document

Github link: https://github.com/keneelshah1/geotab_test

File: Main.Java

1. Void Main()
    a. Added an option to quit

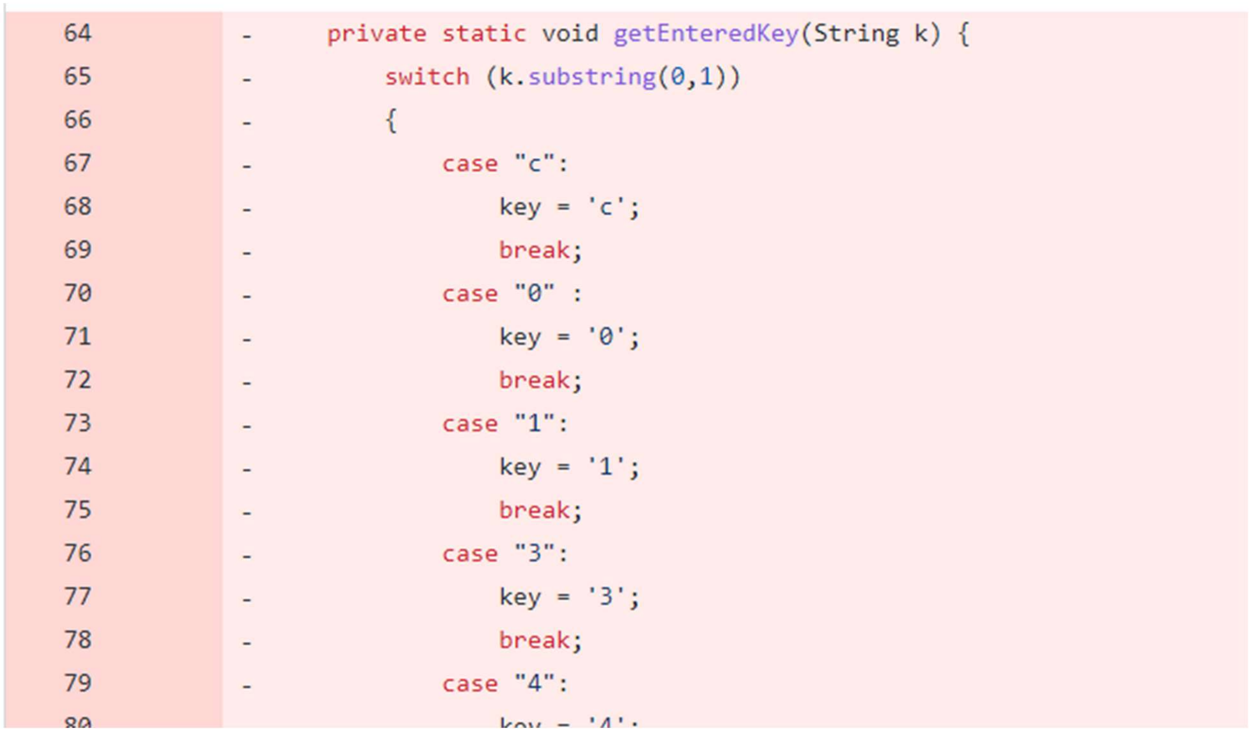
    added a new feature to allow user to quit from the loop.



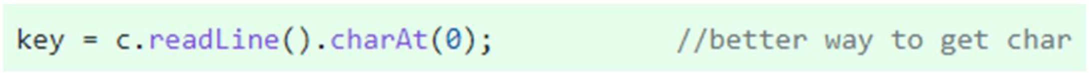
```
+                   printer.Value("Press q to quit").toString(); //added to quit
+ //                  getEnteredKey(c.readLine());
+                   key = c.readLine().charAt(0);           //better way to get char
+
+                   if (key == 'q') //added to quit out of loop
+                   {
+                       printer.Value("See you Soon!").toString();
+                       break;
+                   }
```

    b. Performance Optimization

    Removed the getEnteredKey(String k) function preforming Switch Case. Instead to get the characters used charAt() function. So, instead of calling the function every time this can be used to improve performance.



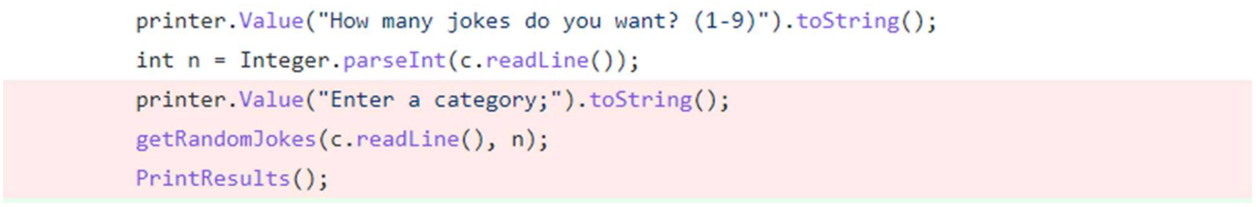
```
-     private static void getEnteredKey(String k) {
-         switch (k.substring(0,1))
-         {
-             case "c":
-                 key = 'c';
-                 break;
-             case "0" :
-                 key = '0';
-                 break;
-             case "1":
-                 key = '1';
-                 break;
-             case "3":
-                 key = '3';
-                 break;
-             case "4":
-                 key = '4';
```



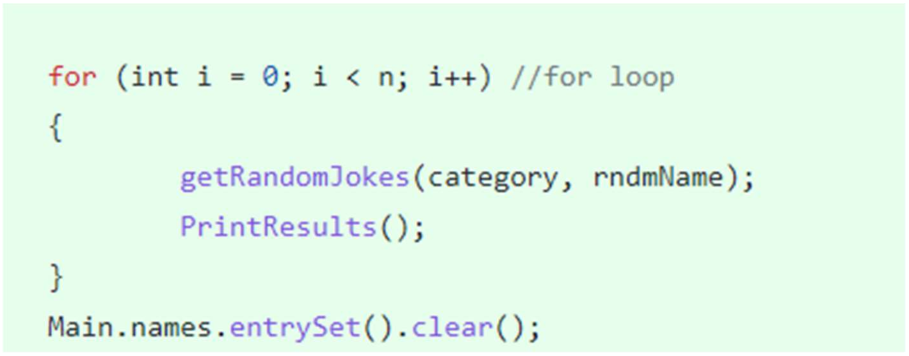
```
key = c.readLine().charAt(0);           //better way to get char
```

c. Added the for loop to print jokes n number of times

Before the code was not able to print the joke more than 1 number of times



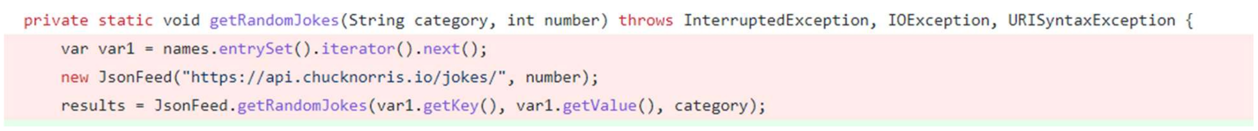
```
printer.Value("How many jokes do you want? (1-9)").toString();
int n = Integer.parseInt(c.readLine());
printer.Value("Enter a category;").toString();
getRandomJokes(c.readLine(), n);
PrintResults();
```



```
for (int i = 0; i < n; i++) //for loop
{
        getRandomJokes(category, rndmName);
        PrintResults();
}
Main.names.entrySet().clear();
```

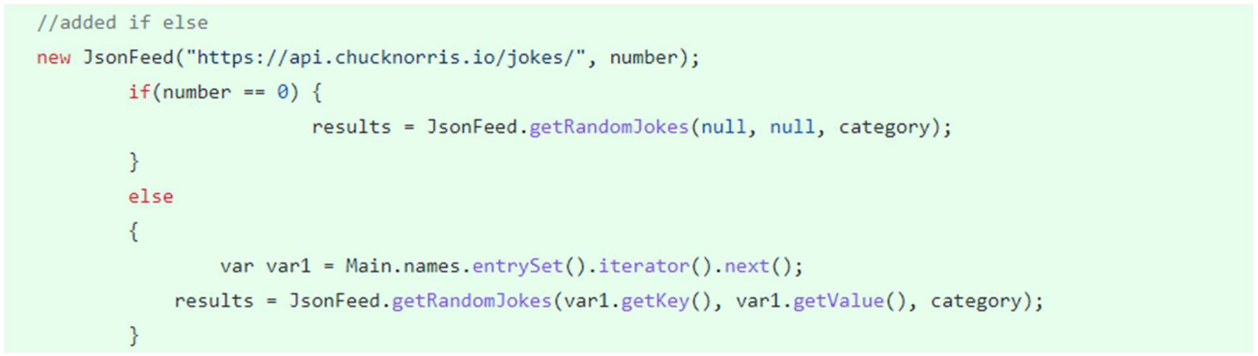
2. getRandomJokes(String category, int number)
    a. extracting values from HashMap

This code was failing when and throwing NullPointerException when called without random names because it tries to get the names from the hash map which is empty.



```
private static void getRandomJokes(String category, int number) throws InterruptedException, IOException, URISyntaxException {
    var var1 = names.entrySet().iterator().next();
    new JsonFeed("https://api.chucknorris.io/jokes/", number);
    results = JsonFeed.getRandomJokes(var1.getKey(), var1.getValue(), category);
```

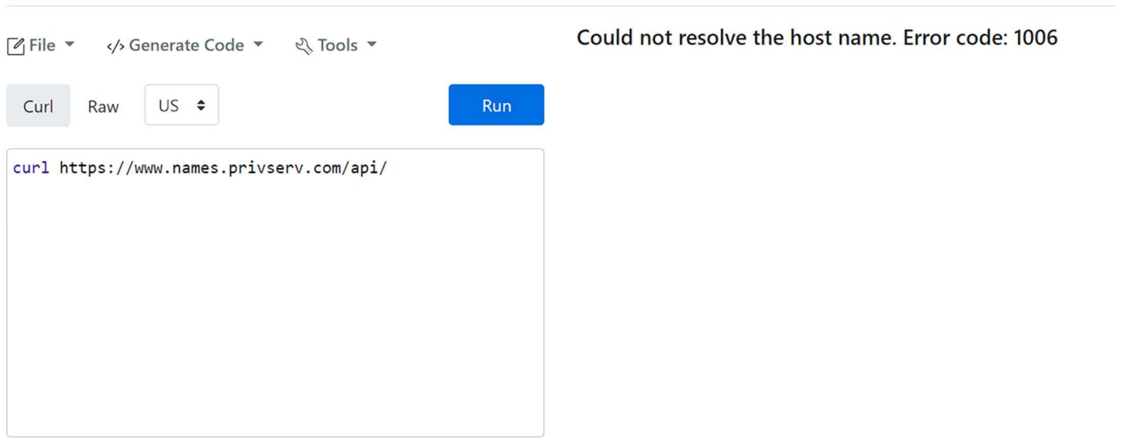
To solve this I added a flag variable ‘rndmName’ if this variable is true it will get the values from hash map and send it to the getRandomJokes() function of JsonFeed Class, else it will send null in names.



```
//added if else
new JsonFeed("https://api.chucknorris.io/jokes/", number);
        if(number == 0) {
                        results = JsonFeed.getRandomJokes(null, null, category);
        }
        else
        {
                var var1 = Main.names.entrySet().iterator().next();
            results = JsonFeed.getRandomJokes(var1.getKey(), var1.getValue(), category);
        }
```

3. Non-working API

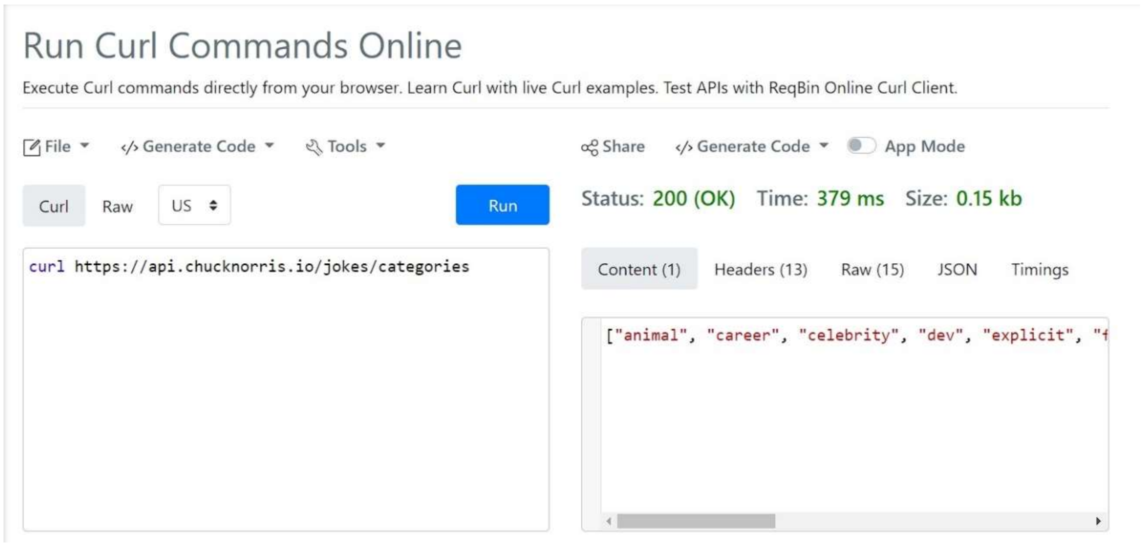
The API to get random names was not working (Host cannot be reached).



```
curl https://www.names.privserv.com/api/
```

Could not resolve the host name. Error code: 1006

However, the API to get the random jokes was working



Run Curl Commands Online

Execute Curl commands directly from your browser. Learn Curl with live Curl examples. Test APIs with ReqBin Online Curl Client.

```
curl https://api.chucknorris.io/jokes/categories
```

Status: 200 (OK) Time: 379 ms Size: 0.15 kb

```
["animal", "career", "celebrity", "dev", "explicit", "f
```

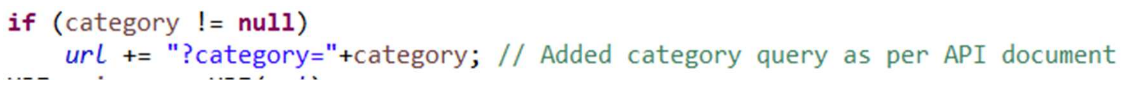
I replaced the API with another API and added a try catch for exception handling to prevent code from crashing.

```
    private static void getNames() throws InterruptedException, IOException, UR
ISyntaxException {
        new JsonFeed("https://api.namefake.com/english-united-states/male/",
0); // wrong domain name please check the api
        Dto dto = JsonFeed.getnames();
        try {
                Main.names.put(dto.getName(), dto.getSurname());
        }
        catch(Exception e)
        {
                e.printStackTrace();
        }
    }
```

File: JsonFeed.java

1. getRandomJokes(String firstname, String lastname, String category)
   a. To get Categories

   As per API documentation the correct way to get jokes from a specific category was to give it as query to updated the code to make the working API link.

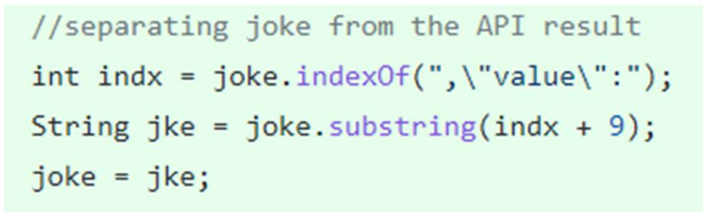


```
if (category != null)
    url += "?category="+category; // Added category query as per API document
```

   There was no functionality to separate the jokes from the API so using substring() method of java extracted the joke from the API result.

   b. Separating joke

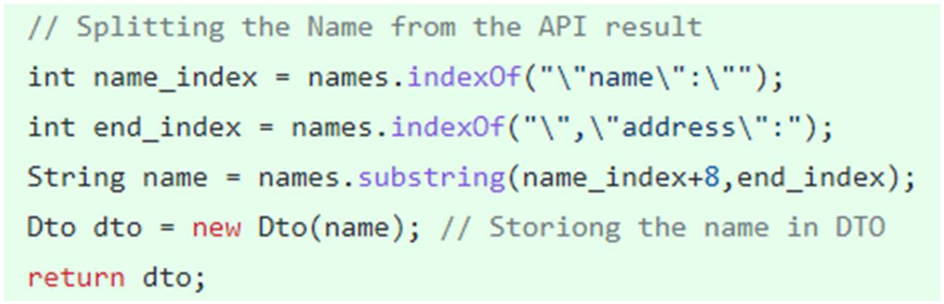
   Added functionality to separate joke from the result fetched from the API and store it into joke variable.



```
//separating joke from the API result
int indx = joke.indexOf(",\"value\":");
String jke = joke.substring(indx + 9);
joke = jke;
```

2. Dto getNames()

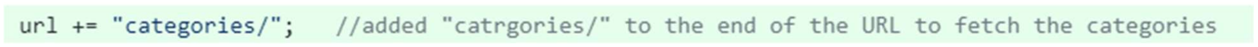
   Added functionality to extract name from the result given by new API and stored it into DTO class object.



```
// Splitting the Name from the API result
int name_index = names.indexOf("\"name\":\"");
int end_index = names.indexOf("\",\"address\":");
String name = names.substring(name_index+8,end_index);
Dto dto = new Dto(name); // Storiong the name in DTO
return dto;
```

3. getCategories()

   a. To get categories
      Updated the URL and appended “categories/” at the end of URL to fetch categories.

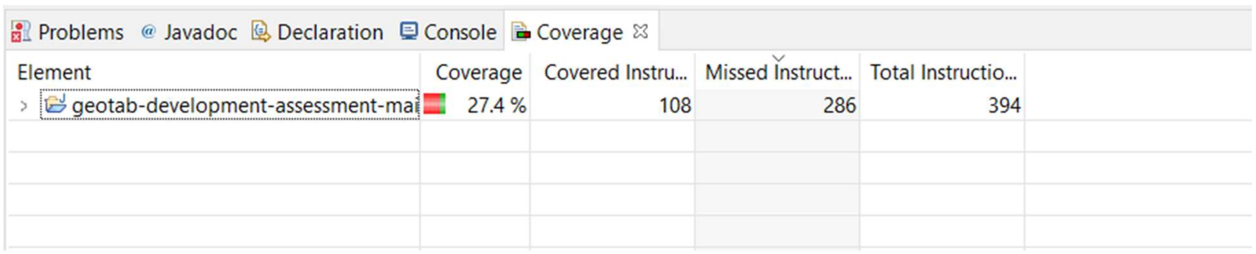


```
url += "categories/";   //added "catrgories/" to the end of the URL to fetch the categories
```

Summary/comments:

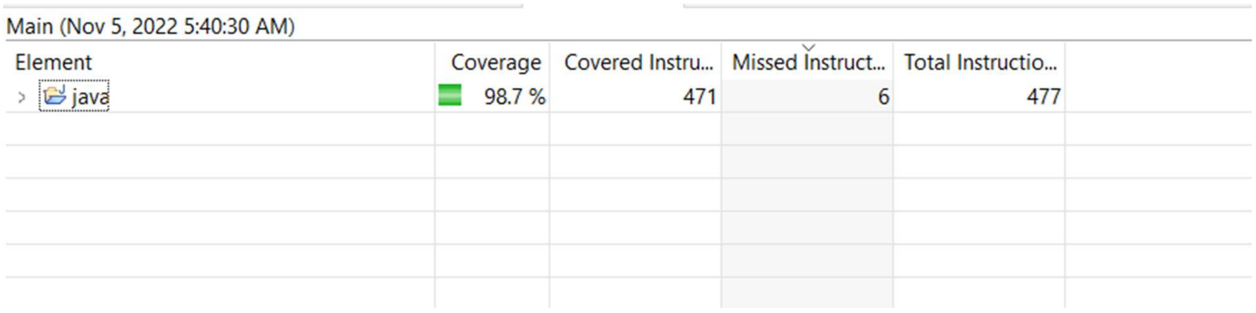
- Readability: Readability of the code was low due to lack of comments, but the variable names and function names were well thought and were easy to understand
- Testability: The code was easy to test and was well separated into function and classes
- Performance: The code was in the infinite loop and there was no way to get out of the loop. Next time give a way to exit an infinite loop.
- Error Handling: Use try catch to handle errors and maintain desired flow of program even when unexpected event happenes.
- Coverage: After the changed the code coverage improved from 27.4% to 98.7%.

Before:



Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
geotab-development-assessment-mai	🟥 27.4 %	108	286	394

After:



Main (Nov 5, 2022 5:40:30 AM)

Element	Coverage	Covered Instru...	Missed Instruct...	Total Instructio...
java	🟩 98.7 %	471	6	477