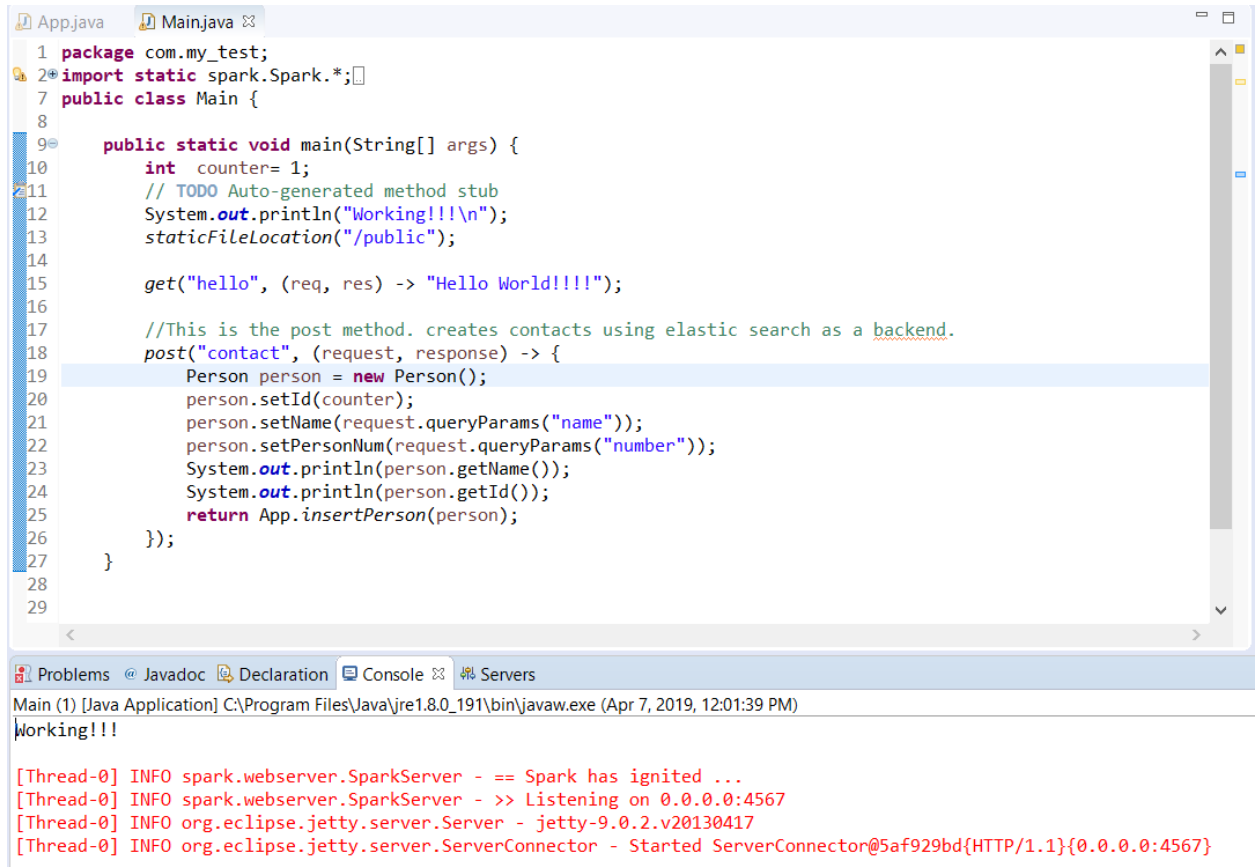


Steps to follow to get started with running to assessment:

1. Download the zip file from the Github repository and import the files as a Maven project in either eclipse or IntelliJ.
2. Download elastic search and run "elasticsearch.bat" file to start the elasticsearch backend.
3. There are 3 class files out of which open "Main.java" class and Run. This will start the sparkjava server.



The screenshot shows an IDE with two tabs: 'App.java' and 'Main.java'. The 'Main.java' tab is active, displaying the following code:

```
1 package com.my_test;
2 import static spark.Spark.*;
7 public class Main {
8
9     public static void main(String[] args) {
10         int counter= 1;
11         // TODO Auto-generated method stub
12         System.out.println("Working!!!\n");
13         staticFileLocation("/public");
14
15         get("hello", (req, res) -> "Hello World!!!");
16
17         //This is the post method. creates contacts using elastic search as a backend.
18         post("contact", (request, response) -> {
19             Person person = new Person();
20             person.setId(counter);
21             person.setName(request.queryParams("name"));
22             person.setPersonNum(request.queryParams("number"));
23             System.out.println(person.getName());
24             System.out.println(person.getId());
25             return App.insertPerson(person);
26         });
27     }
28
29 }
```

Below the code editor, the 'Console' tab is active, showing the following output:

```
Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (Apr 7, 2019, 12:01:39 PM)
Working!!!

[Thread-0] INFO spark.webserver.SparkServer - == Spark has ignited ...
[Thread-0] INFO spark.webserver.SparkServer - >> Listening on 0.0.0.0:4567
[Thread-0] INFO org.eclipse.jetty.server.Server - jetty-9.0.2.v20130417
[Thread-0] INFO org.eclipse.jetty.server.ServerConnector - Started ServerConnector@5af929bd{HTTP/1.1}{0.0.0.0:4567}
```

4. I used postman for sending requests. Send a post request to <http://localhost:4567/contact?name={name}&number={number}> to get data stored in the elasticsearch backend.

(Image on next page).

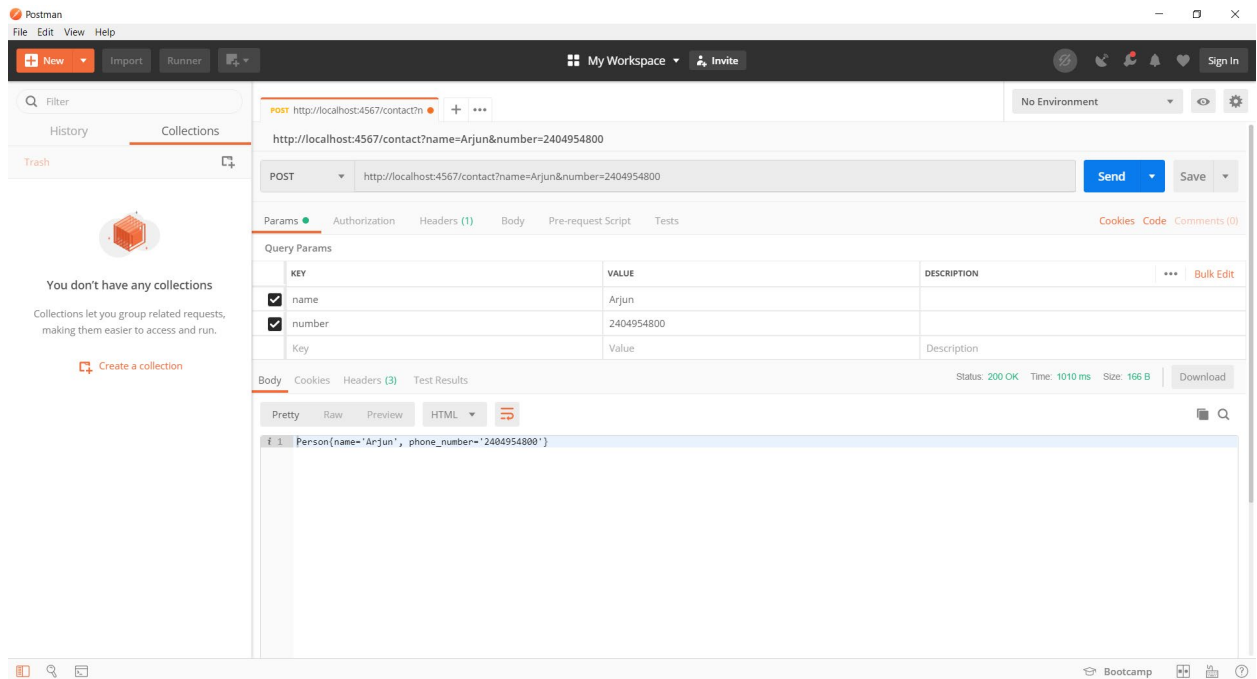


Figure 1- post service working.

5. The “App.java” class is the backend where all the logic developed and the “Main.java” class is the driver. “Person.java” is the model on which the whole application I based. It takes in name and phone number as its private data.