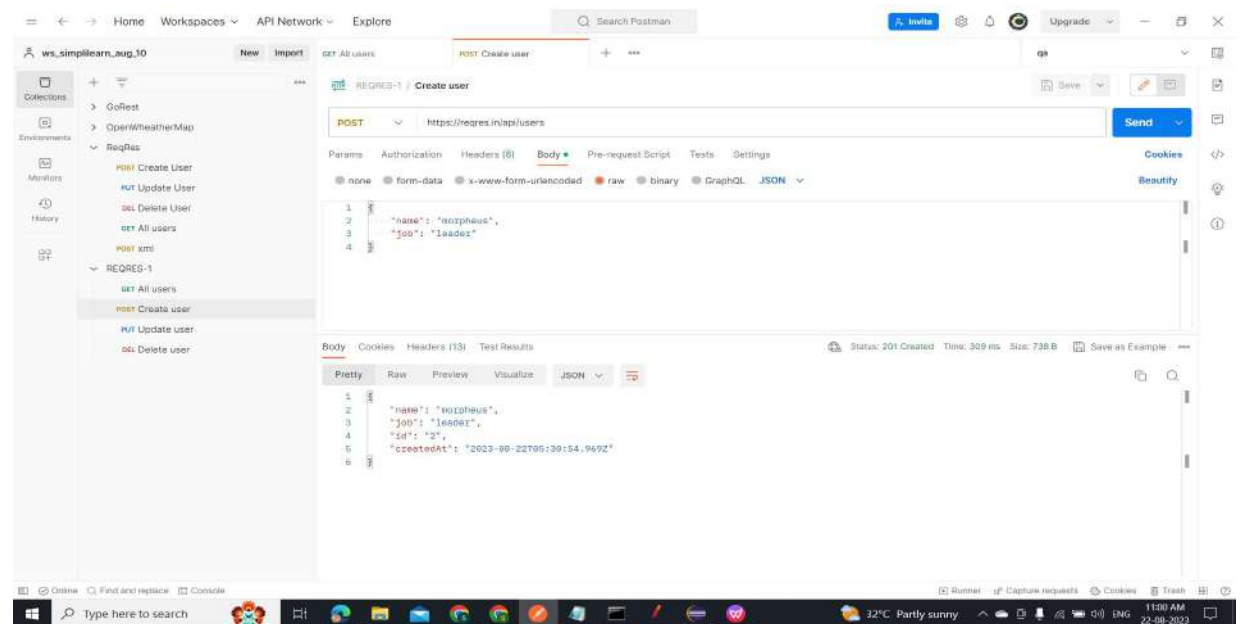
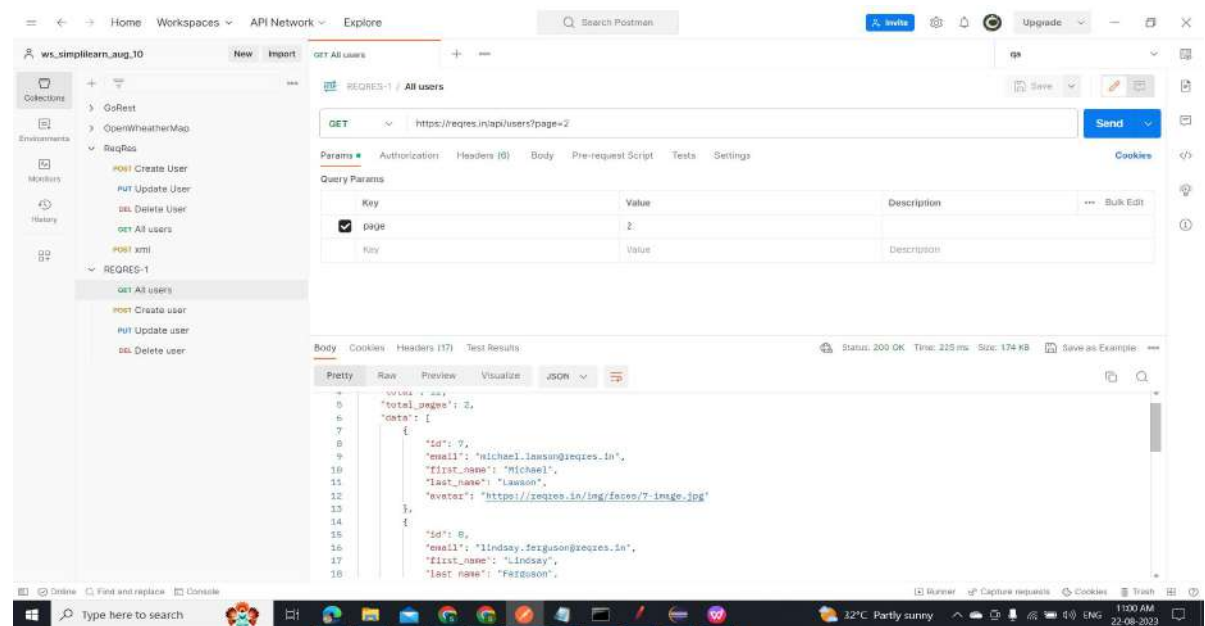
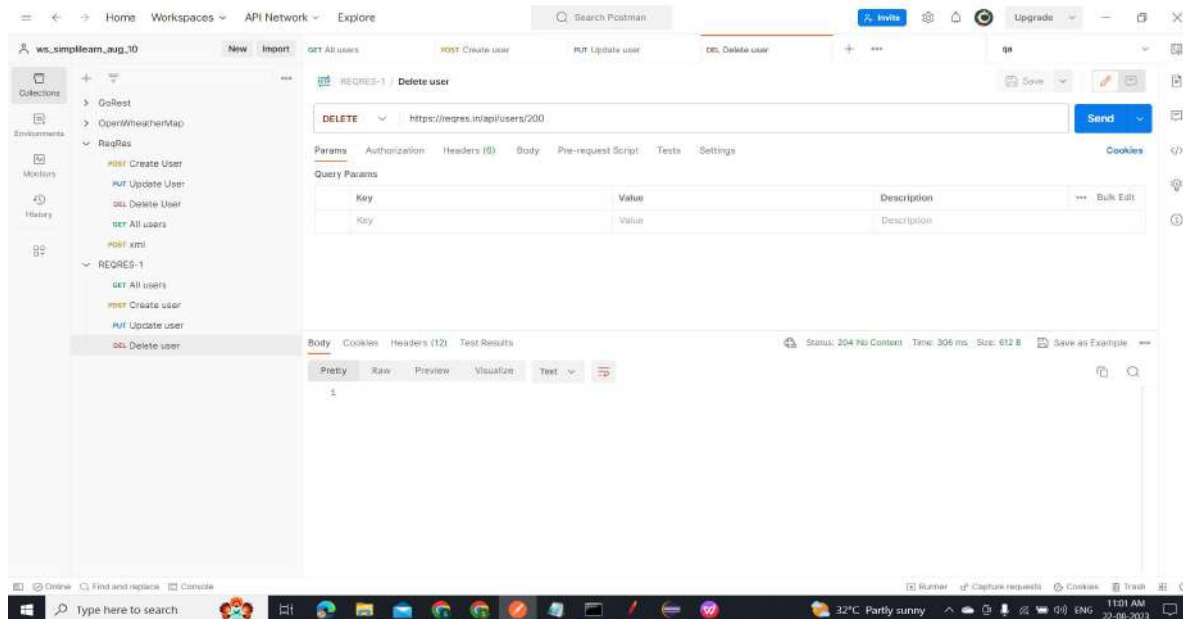
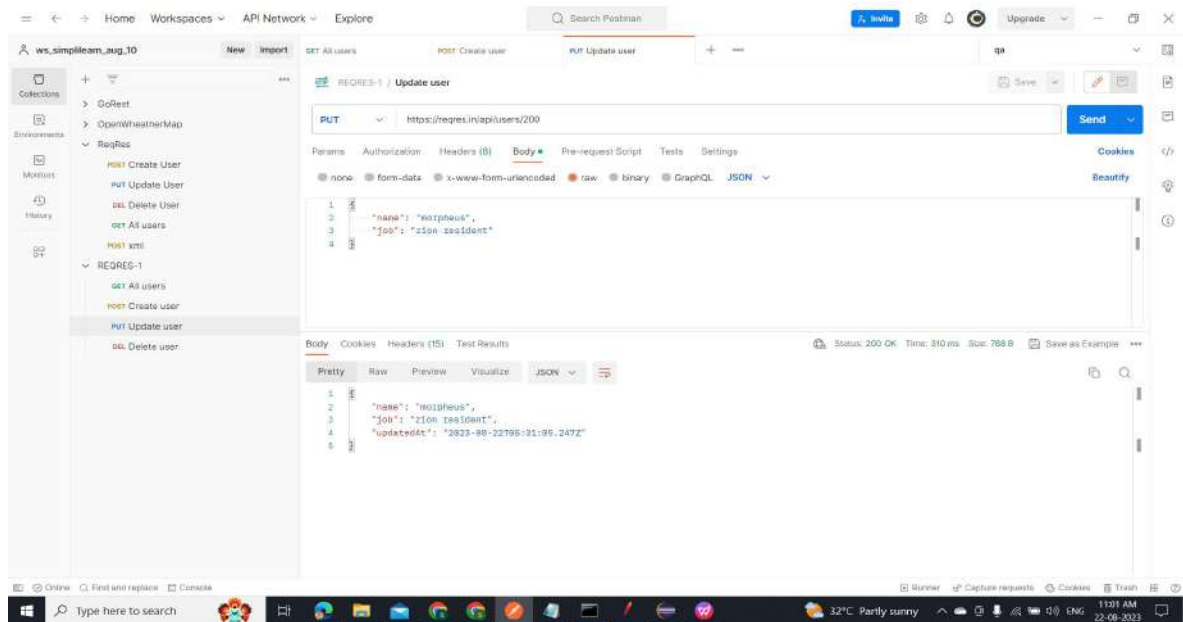


PHASE-3 END PROJECT

BY USING POSTMAN:





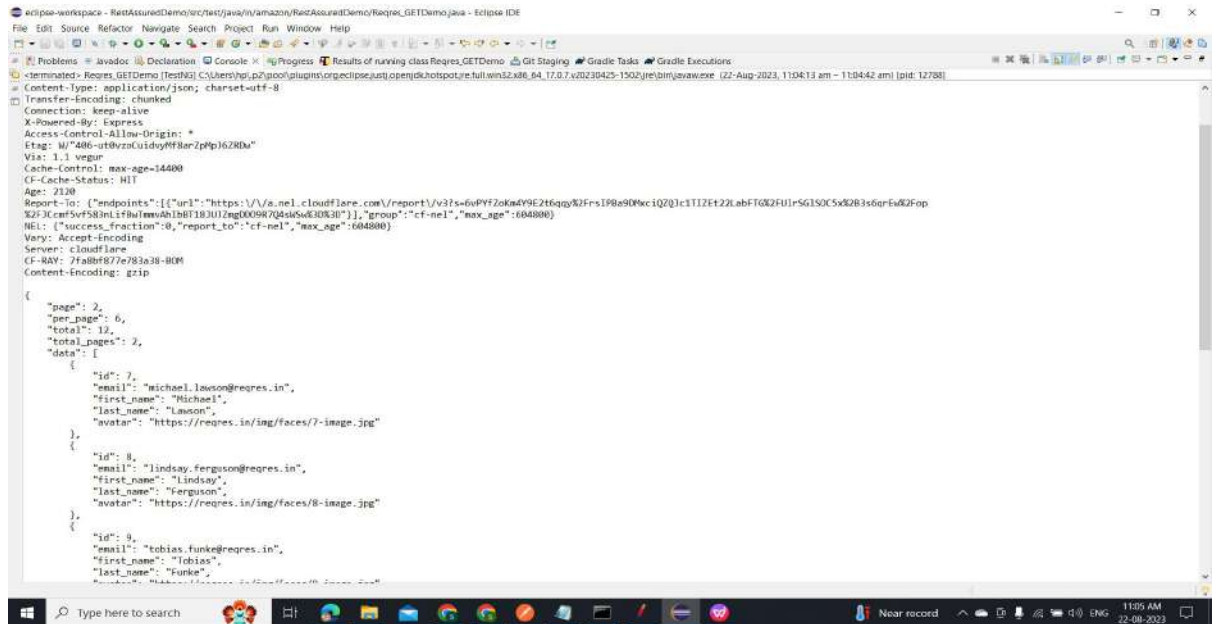
FOR GET REQUEST:

```
package in.amazon.RestAssuredDemo;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class Reqres_GETDemo {
@Test
public void getServerResponse() {
RestAssured
.when()
.get("https://reqres.in/api/users?page=2")
.then()
.assertThat()
```

```

        .statusCode(200)
        .log().all();
    }
}

```

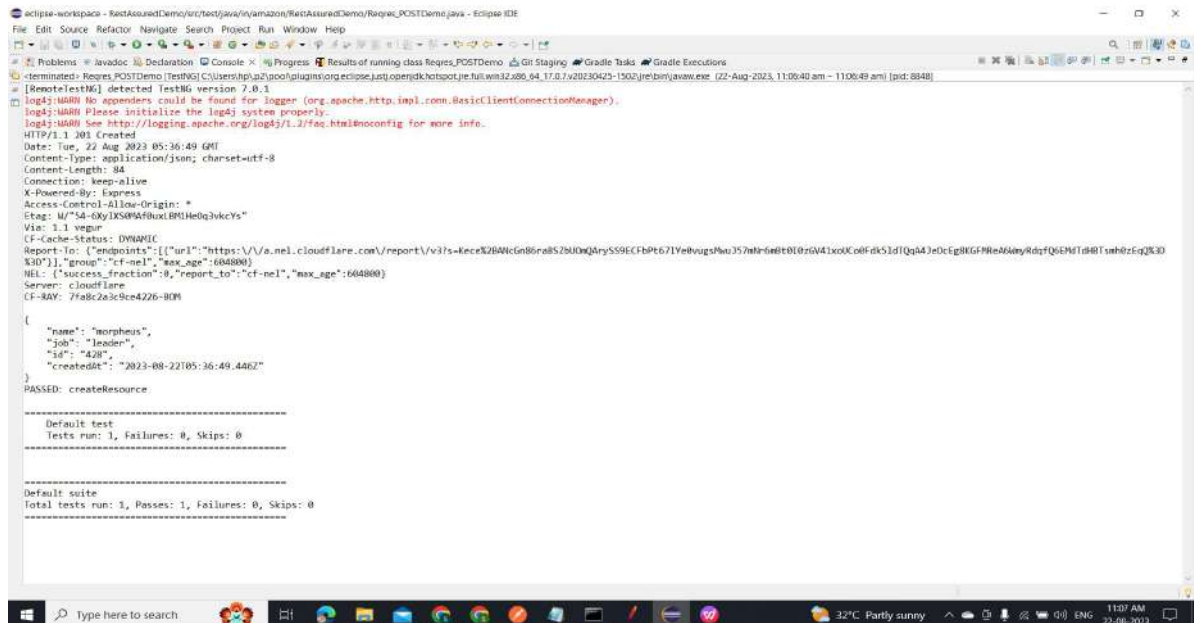


FOR POST REQUEST:

```

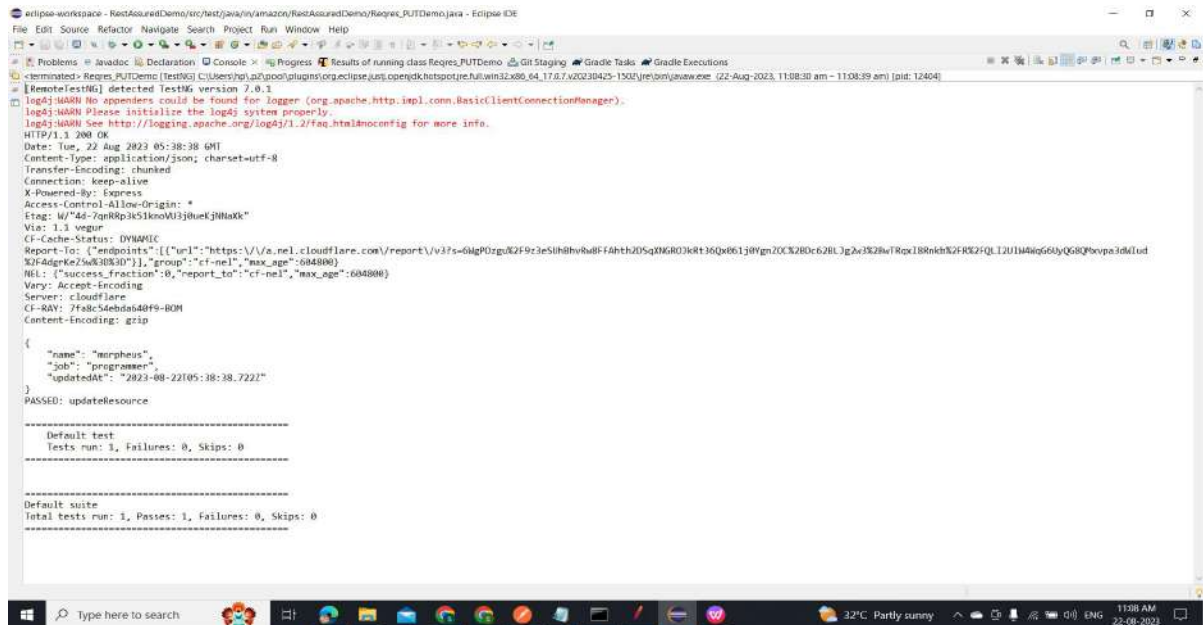
package in.amazon.RestAssuredDemo;
import java.util.HashMap;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class Reqres_POSTDemo {
    HashMap<String, String> map = new HashMap<String, String>();
    @BeforeMethod
    public void createPayload() {
        map.put("name", "morpheus");
        map.put("job", "leader");
        RestAssured.baseURI = "https://reqres.in/";
        RestAssured.basePath = "/api/users";
    }
    @Test
    public void createResource() {
        RestAssured
            .given()
            .contentType("application/json")
            .body(map)
            .when()
            .post()
            .then()
            .statusCode(201)
            .log().all();
    }
}

```



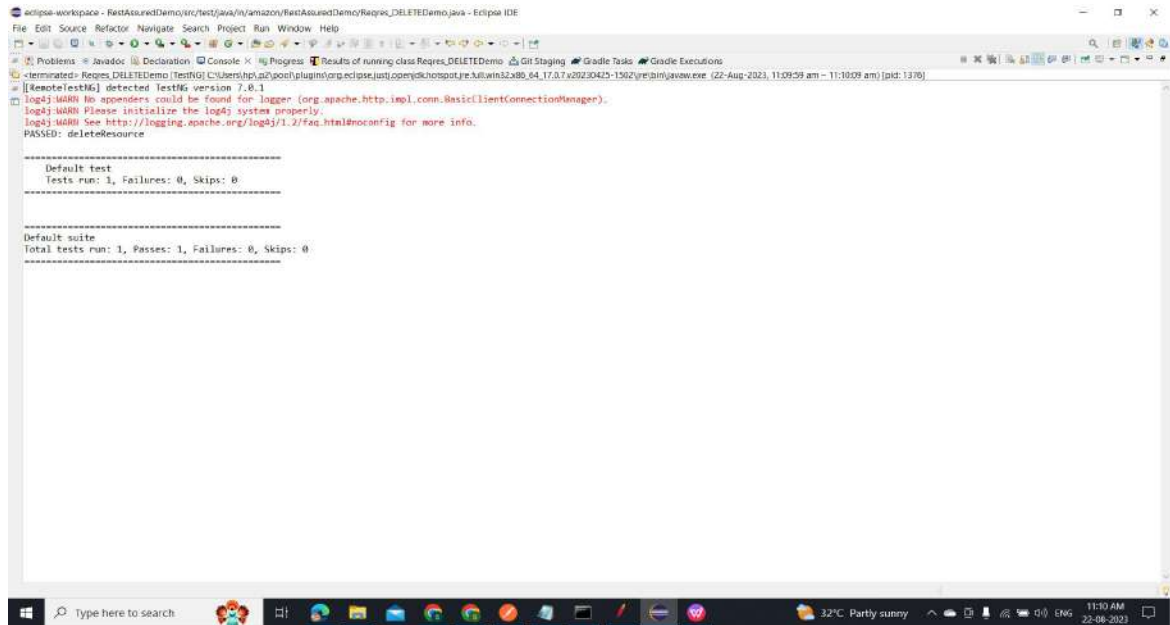
FOR PUT REQUEST:

```
package in.amazon.RestAssuredDemo;
import java.util.HashMap;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class Reqres_PUTDemo {
    HashMap<String, String> map = new HashMap<String, String>();
    @BeforeMethod
    public void createPayload() {
        map.put("name", "morpheus");
        map.put("job", "programmer");
        RestAssured.baseURI = "https://reqres.in/";
        RestAssured.basePath = "/api/users/848";
    }
    @Test
    public void updateResource() {
        RestAssured
            .given()
            .contentType("application/json")
            .body(map)
            .when()
            .put()
            .then()
            .statusCode(200)
            .log().all();
    }
}
```



FOR DELETE REQUEST:

```
package in.amazon.RestAssuredDemo;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class Reqres_DELETEDemo {
@Test
public void deleteResource() {
RestAssured.baseURI = "https://reqres.in/";
RestAssured.basePath = "/api/users/848";
RestAssured
.when()
.delete()
.then()
.statusCode(204);
}
}
```



FOR GET POST PUT DELETE IN ONE CODE:

```

package in.amazon.RestAssuredDemo;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
import java.util.HashMap;
public class Reqres_GET_POST_PUT_DELETEDemo extends TestBase {
    HashMap<String, String> map = new HashMap<String, String>();
    @Test
    public void createPayload() {
        map.put("name", "morpheus");
        map.put("job", "leader");
        RestAssured.baseURI = "https://reqres.in/";
        RestAssured.basePath = "/api/users?page=2";
        RestAssured
            .when()
            .get("/api/users?page=2")
            .then()
            .assertThat()
            .statusCode(200)
            .log().all();
        logger.info("Payload created");
    }
    @Test
    public void createResource() {
        RestAssured
            .given()
            .contentType("application/json")
            .body(map)
            .when()
            .post("/api/users")
            .then()
            .statusCode(201)
            .log().all();
        logger.info("Resource Created Successfully");
    }
    @Test

```

```

public void updateResource() {
    map.put("job", "programmer");
    RestAssured.basePath = "/api/users/848";
    RestAssured
        .given()
        .contentType("application/json")
        .body(map)
        .when()
        .put()
        .then()
        .statusCode(200)
        .log().all();
    logger.info("Resource updated");
}

@Test
public void deleteResource() {
    RestAssured.basePath = "/api/users/848";
    RestAssured
        .when()
        .delete()
        .then()
        .statusCode(204);
    logger.info("Resource Deleted");
}
}

```

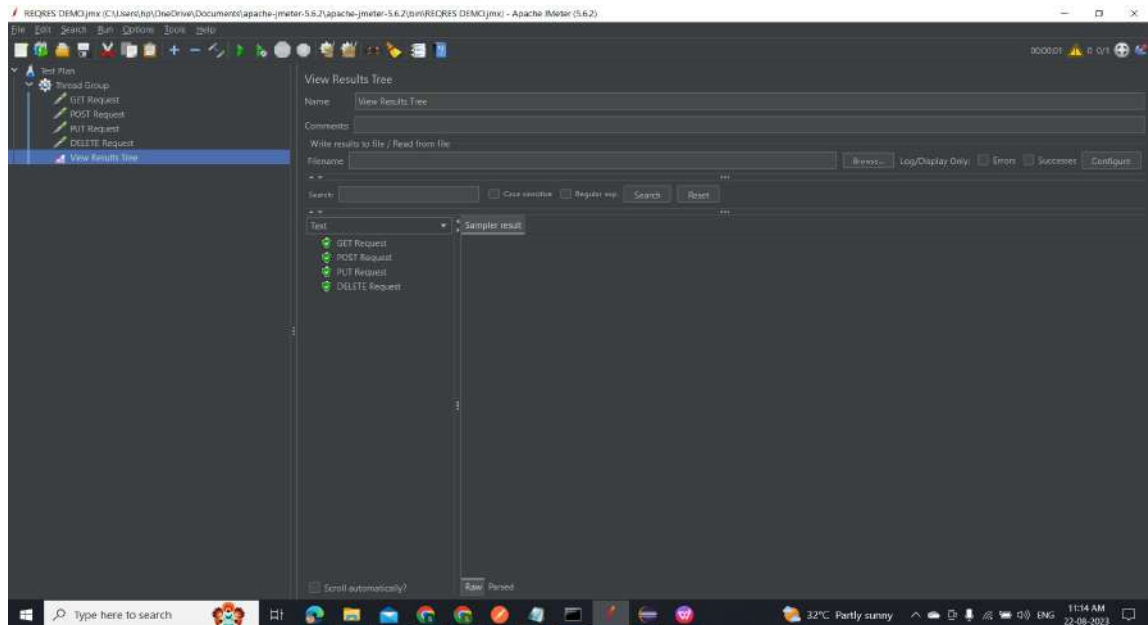
```

INFO 2023-08-22 11:12:16,633 0 LogDemo [main] Payload created
HTTP/1.1 200 OK
Date: Tue, 22 Aug 2023 05:42:17 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 84
Connection: keep-alive
X-Powered-By: Express
Access-Control-Allow-Origin: *
Etag: W/"54-d8sp3jwms6pbobfFTD4gOKbc"
Via: 1.1 vegur
CF-Cache-Status: DYNAMIC
Report-To: [{"endpoints":[{"url":"https://a-nel.cloudflare.com/report/v3?ts=c729427f7782f63100va3i0khpzpxdkk97heY2P053aR8ZG5bJcInnpqz5v55Id55E4571Q8g0K4H70WVaQDTj1866dcL0tr4000ZderJ3AerHgRvdxnzSAK30X807"}], "group":"cf-nel", "max_age":604800}], "success_fraction":0, "report_to":"cf-nel", "max_age":604800}
Server: cloudflare
CF-RAY: 7fabca51d3f3dc-808

{
  "name": "morpheus",
  "job": "leader",
  "id": "477",
  "createdAt": "2023-08-22T05:42:17.344Z"
}
INFO 2023-08-22 11:12:17,363 730 LogDemo [main] Resource Created Successfully
INFO 2023-08-22 11:12:17,794 1161 LogDemo [main] Resource Deleted
HTTP/1.1 204 OK
Date: Tue, 22 Aug 2023 05:42:18 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: Express
Access-Control-Allow-Origin: *
Etag: W/"4d-kKv4q28appEvn7Q/moh/+vx[]"
Via: 1.1 vegur
CF-Cache-Status: DYNAMIC
Report-To: [{"endpoints":[{"url":"https://a-nel.cloudflare.com/report/v3?ts=c729427f7782f63100va3i0khpzpxdkk97heY2P053aR8ZG5bJcInnpqz5v55Id55E4571Q8g0K4H70WVaQDTj1866dcL0tr4000ZderJ3AerHgRvdxnzSAK30X807"}], "group":"cf-nel", "max_age":604800}], "success_fraction":0, "report_to":"cf-nel", "max_age":604800}
Server: cloudflare
CF-RAY: 7fabca51d3f3dc-808

```


BY USING JMETER:



USING BLAZEMETER TESTING:

