



Instituto Politécnico Nacional

Escuela Superior de Cómputo



“Tarea 8. Desarrollo de un cliente para un servicio web REST”

Desarrollo de Sistemas Distribuidos

4CM5

Profesor Pineda Guerrero Carlos

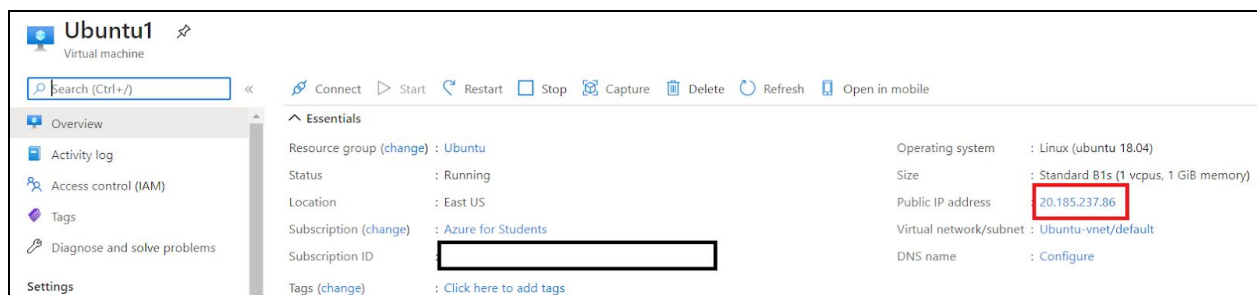
Poblano Díaz Luis Edgar

Desarrollo

Para esta tarea, vamos a implementar un programa que funcione como cliente para el servicio implementado en la Tarea 7. El procedimiento para levantar el servicio va a ser exactamente el mismo, con la única diferencia de que vamos a tener un cambio en uno de los archivos: en la clase *Servicio.java* del archivo *Servicio.zip* en donde implementamos el método para borrar todos los usuarios necesario para esta tarea. La función que agregamos en esta clase es la siguiente:

```
@POST
@Path("/borratodo")
@Consumes(MediaType.APPLICATION_FORM_URLENCODED)
@Produces(MediaType.APPLICATION_JSON)
public Response borrar_usuarios(@FormParam("empty") String s)
throws Exception{
    Connection c = pool.getConnection();
    try{
        PreparedStatement stmt_1 = c.prepareStatement("DELETE FROM
fotos_usuarios");
        try{
            stmt_1.executeUpdate();
        }
        finally{
            stmt_1.close();
        }
        PreparedStatement stmt_2 = c.prepareStatement("DELETE FROM
usuarios");
        try{
            stmt_2.executeUpdate();
        }
        finally{
            stmt_2.close();
        }
    }
    catch(Exception e){
        return Response.status(400).entity(j.toJson(new
Error(e.getMessage()))).build();
    }
    finally{
        c.close();
    }
    return Response.ok().build();
}
```

La máquina virtual en donde va a estar el servicio va a ser la siguiente:



El cliente va a tener las opciones de dar de alta, consultar o borrar un usuario, además de la nueva opción implementada de borrar todos los usuarios, y por último una opción para salir. El código del método *main* va a ser el siguiente, en donde se selecciona cualquiera de estas opciones. (El método menú dentro del ciclo sólo se encarga de imprimir las opciones disponibles).

```
public static void main(String [] args) throws IOException{
    InputStreamReader isr = new InputStreamReader(System.in);
    BufferedReader br = new BufferedReader(isr);
    j = new GsonBuilder().setDateFormat("yyyy-MM-
    dd'T'HH:mm:ss.SSS").create();
    boolean exec = true;
    while(exec){
        menu();
        String opt = br.readLine();
        switch(opt){
            case "a":
                alta(br);
                break;
            case "b":
                consulta(br);
                break;
            case "c":
                borra(br);
                break;
            case "d":
                borra_todo();
                break;
            case "e":
                exec = false;
                break;
            default:
                System.out.println("\nOpción no válida\n");
        }
    }
}
```

La función *alta* va a tener la siguiente forma, en donde primero se obtienen todos los datos del usuario que se va a registrar (en este caso no se va a solicitar ninguna foto, se va a dar de alta con *null*). Después, se prepara la conexión con la dirección del servidor en la que se realizan las operaciones de alta. En este caso, vamos a crear una instancia de la clase *Usuario* y esta se serializa utilizando Gson. Esta cadena es la que se va a mandar como parámetro al servidor y en caso de concretarse de manera exitosa la operación, se va a imprimir el mensaje "OK". En caso contrario, nos va a imprimir el código de error HTTP devuelto por el servidor. A continuación, se muestran las imágenes de la clase *Usuario* con todos sus métodos, y después el código de la función *alta()*.

```

public class Usuario{

    String email;
    String nombre;
    String apellido_paterno;
    String apellido_materno;
    String fecha_nacimiento;
    String telefono;
    String genero;
    byte[] foto;

    public Usuario(String email, String nombre, String apellido_paterno, String apellido_materno,
String fecha_nacimiento, String telefono, String genero, byte[] foto){
        this.email = email;
        this.nombre = nombre;
        this.apellido_paterno = apellido_paterno;
        this.apellido_materno = apellido_materno;
        this.fecha_nacimiento = fecha_nacimiento;
        this.telefono = telefono;
        this.genero = genero;
        this.foto = foto;
    }

    public String toString(){
        return "\nEmail = " + email + "\nNombre = " + nombre + "\nApellido Paterno = " +
apellido_paterno +
        "\nApellido Materno = " + apellido_materno + "\nFecha de Nacimiento = " +
fecha_nacimiento +
        "\nTelefono = " + telefono + "\nGenero = " + genero;
    }
}

public static void alta(BufferedReader br){
    try{
        System.out.println("\nIngresa el email del usuario que deseas dar de alta:");
        String email = br.readLine();
        System.out.println("Ingresa el nombre del usuario:");
        String nombre = br.readLine();
        System.out.println("Ingresa el apellido paterno del usuario:");
        String ap = br.readLine();
        System.out.println("Ingresa el apellido materno del usuario:");
        String am = br.readLine();
        System.out.println("Ingresa la fecha de nacimiento del usuario con formato aaaa-mm-dd:");
        String fecha = br.readLine();
        System.out.println("Ingresa el teléfono del usuario:");
        String tele = br.readLine();
        System.out.println("Ingresa el género del usuario ('M' O 'F'):");
        String gen = br.readLine();
        while(!gen.equals("M") && !gen.equals("F")){
            System.out.println("Ingresa el género del usuario ('M' O 'F'):");
            gen = br.readLine();
        }

        URL url = new URL("http://" + server_ip + ":8080/Servicio/rest/ws/alta");
        HttpURLConnection huc = (HttpURLConnection) url.openConnection();
        huc.setDoOutput(true);
        huc.setRequestMethod("POST");
        huc.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");

        Usuario u = new Usuario(email,nombre,ap,am,fecha,tele,gen,null);
        String params = "usuario=" + j.toJson(u);
        OutputStream os = huc.getOutputStream();
        os.write(params.getBytes());
        os.flush();
        if (huc.getResponseCode() != HttpURLConnection.HTTP_OK){
            System.out.println("Codigo de error HTTP: " + huc.getResponseCode());
            return ;
        }
        System.out.println("\nOK");
        huc.disconnect();
    }
    catch(IOException ie){ System.out.println("\nError en el alta\n"); }
}

```

Después, vamos a ver el método *consulta()* donde se solicita el email del usuario que se quiere obtener. Luego se prepara la conexión con la URL correspondiente al método de consulta del servidor. En este caso, el parámetro enviado es el email, por lo que no se tiene que serializar nada antes. En caso de que haya un error, muestra el código del mismo y se sale. En caso contrario, se obtiene una cadena desde el servidor (la cual

está en formato de Json). Vamos a utilizar Gson para deserializar esta cadena y convertirla en una instancia de la clase *Usuario*, la cual se va a imprimir con el método *toString* definido en la propia clase. El código de este método es el siguiente:

```
public static void consulta(BufferedReader br){
    try{
        System.out.println("\nIngresa el email del usuario que deseas consultar:");
        String email = br.readLine();

        URL url = new URL("http://" + server_ip + ":8080/Servicio/rest/ws/consulta");
        HttpURLConnection huc = (HttpURLConnection) url.openConnection();
        huc.setDoOutput(true);
        huc.setRequestMethod("POST");
        huc.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");

        String params = "email=" + URLEncoder.encode(email, "UTF-8");
        OutputStream os = huc.getOutputStream();
        os.write(params.getBytes());
        os.flush();
        if (huc.getResponseCode() != HttpURLConnection.HTTP_OK){
            System.out.println("Codigo de error HTTP: " + huc.getResponseCode());
            return ;
        }

        BufferedReader br2 = new BufferedReader(new InputStreamReader((huc.getInputStream())));
        String ans;
        Usuario u = null;
        while((ans = br2.readLine()) != null)
            u = (Usuario)j.fromJson(ans, Usuario.class);
        System.out.println(u);
        huc.disconnect();
    }
    catch(IOException ie){ System.out.println("\nError en la consulta\n"); }
}
```

Ahora, para el método *borra()*, vamos a tener casi lo mismo que en el método anterior, con la diferencia de que ahora el servidor no devolverá una string y que tenemos que acceder a la URL correspondiente al método *borra* del servidor. Si la operación es exitosa, se imprime la cadena "OK", en caso contrario se imprime el código del error. El código de este método es el siguiente:

```
public static void borra(BufferedReader br){
    try{
        System.out.println("\nIngresa el email del usuario que deseas borrar:");
        String email = br.readLine();

        URL url = new URL("http://" + server_ip + ":8080/Servicio/rest/ws/borra");
        HttpURLConnection huc = (HttpURLConnection) url.openConnection();
        huc.setDoOutput(true);
        huc.setRequestMethod("POST");
        huc.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");

        String params = "email=" + URLEncoder.encode(email, "UTF-8");
        OutputStream os = huc.getOutputStream();
        os.write(params.getBytes());
        os.flush();
        if (huc.getResponseCode() != HttpURLConnection.HTTP_OK){
            System.out.println("Codigo de error HTTP: " + huc.getResponseCode());
            return ;
        }
        System.out.println("\nOK");
        huc.disconnect();
    }
    catch(IOException ie){ System.out.println("\nError al borrar\n"); }
}
```

Por último, vamos a ver el método *borra_todo()* que borra todos los usuarios dentro de la base de datos. Este método es prácticamente el mismo que el método *borra()* con la

diferencia de que ya no es necesario especificar ningún email, el URL correspondiente al método, y en parámetros sólo vamos a mandar la cadena "empty=" por cómo especificamos el método en el servidor.

```
public static void borra_todo(){
    try{
        URL url = new URL("http://" + server_ip + ":8080/Servicio/rest/ws/borratodo");
        HttpURLConnection huc = (HttpURLConnection) url.openConnection();
        huc.setDoOutput(true);
        huc.setRequestMethod("POST");
        huc.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");

        String params = "empty=";
        OutputStream os = huc.getOutputStream();
        os.write(params.getBytes());
        os.flush();
        if (huc.getResponseCode() != HttpURLConnection.HTTP_OK){
            System.out.println("Codigo de error HTTP: " + huc.getResponseCode());
            return ;
        }
        System.out.println("\nOK");
        huc.disconnect();
    }
    catch(IOException ie){ System.out.println("\nError al borrar\n"); }
}
```

Cabe aclarar que hay dos atributos estáticos de la clase para poder controlar más fácil cuando se requiera cambiar la ip del servidor y para no enviar el objeto de Gson en todos los métodos.

```
public static String server_ip = "20.185.237.86";
public static Gson j;
```

Pruebas

Para compilar y ejecutar el programa, tenemos que hacerlo de la siguiente forma (además de tener el .jar de gson necesario para utilizar la clase).

```
edgar@edgar-VirtualBox:~/shared/7 Semestre/Distribuidos/Tarea8$ javac -cp gson-2.8.6.jar *.java
edgar@edgar-VirtualBox:~/shared/7 Semestre/Distribuidos/Tarea8$ java -cp gson-2.8.6.jar:. Client

MENU

a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir

Opción: █
```

Este será el guión que vamos a realizar para mostrar que funciona correctamente nuestro programa.

1. Consultar los correos luis@gmail.com, pancho@hotmail.com, juanito@yahoo.com.mx. Aquí veremos que nos devuelve el error 400 debido a que ninguno de los usuarios está registrado.
2. Dar de alta los 3 correos y volverlos a consultar.
3. Borrar el usuario con correo luis@gmail.com y consultarlo.
4. Borrar todos los usuarios y consultar los otros dos correos restantes.
5. Salir.

```
edgar@edgar-VirtualBox:~/shared/7 Semestre/Distribuidos/Tareas$ javac -cp gson-2.8.6.jar *.java
edgar@edgar-VirtualBox:~/shared/7 Semestre/Distribuidos/Tareas$ java -cp gson-2.8.6.jar:. Client
```

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: b

Ingresa el email del usuario que deseas consultar:

luis@gmail.com

Codigo de error HTTP: 400

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: b

Ingresa el email del usuario que deseas consultar:

pancho@hotmail.com

Codigo de error HTTP: 400

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: b

Ingresa el email del usuario que deseas consultar:

juanito@yahoo.com.mx

Codigo de error HTTP: 400

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: a

Ingresa el email del usuario que deseas dar de alta:

luis@gmail.com

Ingresa el nombre del usuario:

Luis

Ingresa el apellido paterno del usuario:

Poblano

Ingresa el apellido materno del usuario:

Diaz

Ingresa la fecha de nacimiento del usuario con formato aaaa-mm-dd:

1999-10-21

Ingresa el teléfono del usuario:

5512345678

Ingresa el género del usuario ('M' o 'F'):

M

OK


```
MENU

a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir

Opción: a

Ingresa el email del usuario que deseas dar de alta:
pancho@hotmail.com
Ingresa el nombre del usuario:
Pancho
Ingresa el apellido paterno del usuario:
Perez
Ingresa el apellido materno del usuario:
Perez
Ingresa la fecha de nacimiento del usuario con formato aaaa-mm-dd:
1985-07-15
Ingresa el teléfono del usuario:
5555555555
Ingresa el género del usuario ('M' O 'F'):
M

OK
```

```
MENU

a. Alta usuario
b. Consulta usuario
c. Borra usuario
d. Borra todos los usuarios
e. Salir

Opción: a

Ingresa el email del usuario que deseas dar de alta:
juanito@yahoo.com.mx
Ingresa el nombre del usuario:
Juan
Ingresa el apellido paterno del usuario:
Jimenez
Ingresa el apellido materno del usuario:
Mendez
Ingresa la fecha de nacimiento del usuario con formato aaaa-mm-dd:
2002-12-31
Ingresa el teléfono del usuario:
5500112233
Ingresa el género del usuario ('M' O 'F'):
M

OK
```

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: b

Ingresa el email del usuario que deseas consultar:
luis@gmail.com

Email = luis@gmail.com
Nombre = Luis
Apellido Paterno = Poblano
Apellido Materno = Diaz
Fecha de Nacimiento = 1999-10-21
Telefono = 5512345678
Genero = M

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: b

Ingresa el email del usuario que deseas consultar:
pancho@hotmail.com

Email = pancho@hotmail.com
Nombre = Pancho
Apellido Paterno = Perez
Apellido Materno = Perez
Fecha de Nacimiento = 1985-07-15
Telefono = 5555555555
Genero = M

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: b

Ingresa el email del usuario que deseas consultar:
juanito@yahoo.com.mx

Email = juanito@yahoo.com.mx
Nombre = Juan
Apellido Paterno = Jimenez
Apellido Materno = Mendez
Fecha de Nacimiento = 2002-12-31
Telefono = 5500112233
Genero = M

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: c

Ingresa el email del usuario que deseas borrar:
luis@gmail.com

OK

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: b

Ingresa el email del usuario que deseas consultar:
luis@gmail.com
Codigo de error HTTP: 400

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: d

OK

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: b

Ingresa el email del usuario que deseas consultar:

pancho@hotmail.com

Codigo de error HTTP: 400

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: b

Ingresa el email del usuario que deseas consultar:

juanito@yahoo.com.mx

Codigo de error HTTP: 400

MENU

- a. Alta usuario
- b. Consulta usuario
- c. Borra usuario
- d. Borra todos los usuarios
- e. Salir

Opción: e

edgar@edgar-VirtualBox:~/shared/7 Semestre/Distribuidos/Tarea8\$

Conclusiones

Con esta práctica, pudimos ver cómo se consume una aplicación mediante una aplicación, en lugar de un navegador web. Con lo que se podrá abstraer en un futuro para desarrollar aplicaciones que envíen y reciban recursos con un servidor utilizando las herramientas para interactuar con objetos de tipo Json. También, observamos que añadir nuevos métodos para realizar otras operaciones es sencillo ya una vez teniendo un producto como el que teníamos de la tarea anterior.