

## Explicación API Rest

### Entidad sensor\_valor:

#### Metodo Get →

-Url: localhost:8080/api/sensor\_valor/:id\_sensor → donde :id\_sensor es la id del sensor del cual queremos hacer la consulta

#### -Metodo:

```
private void get_sensor_valor(RoutingContext routingContext) {  
    MySQLPool.query("SELECT * FROM daddatabase.sensor_valor WHERE id_sensor = "  
        + routingContext.request().getParam("id_sensor"), res -> {  
        if (res.succeeded()) {  
            RowSet<Row> resultSet = res.result();  
            System.out.println("Consulta satisfactoria");  
            JSONArray result = new JSONArray();  
            for (Row row : resultSet) {  
                result.add(JSONObject.mapFrom(new sensor_valor(row.getInteger("id_sensor_valor"),  
                    row.getInteger("id_sensor"), row.getFloat("valor"), row.getFloat("precision_valor"),  
                    row.getLong("tiempo"))));  
            }  
            routingContext.response().setStatusCode(200).putHeader("content-type", "application/json")  
                .end(result.encodePrettily());  
        } else {  
            System.out.println("Consulta fallida");  
            routingContext.response().setStatusCode(401).putHeader("content-type", "application/json")  
                .end(JSONObject.mapFrom(res.cause()).encodePrettily());  
        }  
    });  
}
```

En el cual se hace una query con un SELECT donde se le pasara la id del sensor, y, si la consulta es satisfactoria, se nos proporcionará un objetos JSONArray con todos los Json que contengan la información de la BBDD. En caso de que sea fallida, se mostrará un mensaje de error.

#### -Captura Postman:

The screenshot shows a Postman interface for a GET request to `localhost:8080/api/sensor_valor/1`. The request is successful, returning a 200 OK status with a response time of 864 ms. The response body is a JSON array containing three objects, each representing a sensor reading with fields like id\_sensor\_valor, id\_sensor, valor, precision\_valor, and tiempo.

```
[{"id_sensor_valor": 1, "id_sensor": 1, "valor": 32.0, "precision_valor": 2.0, "tiempo": 122345654}, {"id_sensor_valor": 10, "id_sensor": 1, "valor": 40.0, "precision_valor": 3.0, "tiempo": 1234567654}, {"id_sensor_valor": 11, "id_sensor": 1, "valor": 40.0, "precision_valor": 3.0, "tiempo": 1234567654}, {"id_sensor_valor": 12, "id_sensor": 1, "valor": 40.0, "precision_valor": 3.0, "tiempo": 1234567654}]
```

## Metodo Put:

-Url: localhost:8080/api/sensor\_valor

### -Metodo:

```
private void put_sensor_valor(RoutingContext routingContext) {
    sensor_valor sensor_valor = Json.decodeValue(routingContext.getBodyAsString(), sensor_valor.class);
    MySQLPool.preparedStatement(
        "INSERT INTO daddatabase.sensor_valor (id_sensor, valor, precision_valor, tiempo) VALUES (?, ?, ?, ?)",
        Tuple.of(sensor_valor.getId_sensor(), sensor_valor.getValor(), sensor_valor.getPrecision_valor(),
            sensor_valor.getTiempo()),
        handler -> {
            if (handler.succeeded()) {
                System.out.println("Añadida correctamente");
                System.out.println(handler.result().rowCount());

                long id = handler.result().property(MySQLClient.LAST_INSERTED_ID);
                sensor_valor.setId_sensor_valor((int) id);

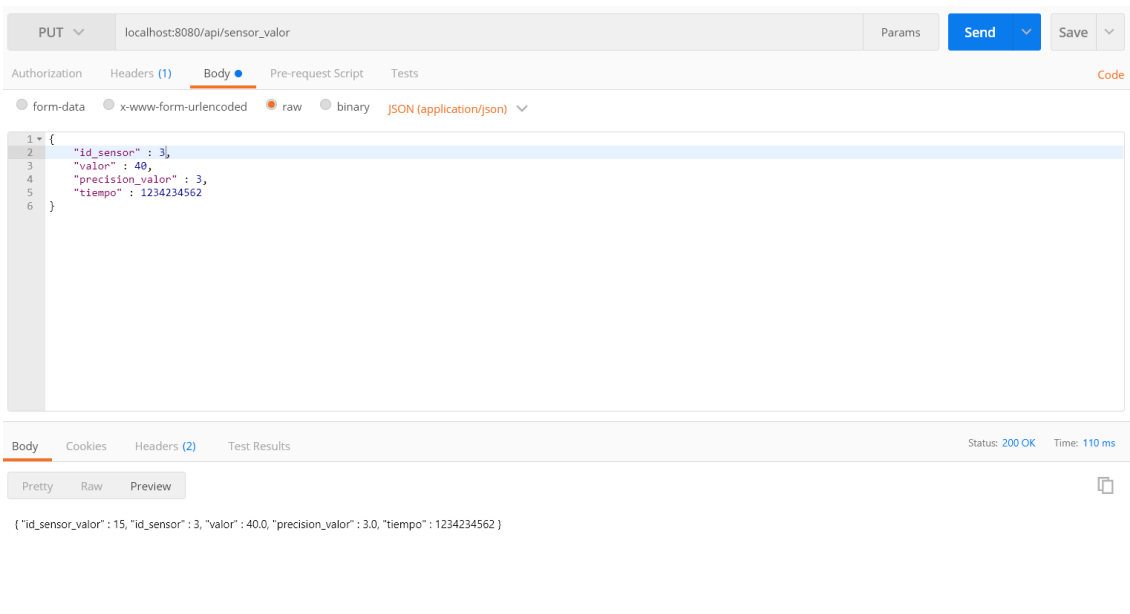
                routingContext.response().setStatusCode(200).putHeader("content-type", "application/json")
                    .end(JsonObject.mapFrom(sensor_valor).encodePretty());
            } else {
                System.out.println("Algo salió mal");
                System.out.println(handler.cause().toString());
                routingContext.response().setStatusCode(401).putHeader("content-type", "application/json")
                    .end(JsonObject.mapFrom(handler.cause()).encodePretty());
            }
        });
}
```

En el cual se hace una query con un INSERT donde se le pasara los valores de los parámetros que se piden en forma de objeto Json, y, si la prueba es satisfactoria, se añadirá a la información de la BBDD. En caso de que sea fallida, se mostrará un mensaje de error.

Jason a proporcionar por el cliente:

```
{
  "id_sensor" : 1,
  "valor" : 40,
  "precisión_valor" : 3,
  "tiempo" : 1234234562
}
```

### -Captura Postman:



## Entidad sensor:

### Metodo Get →

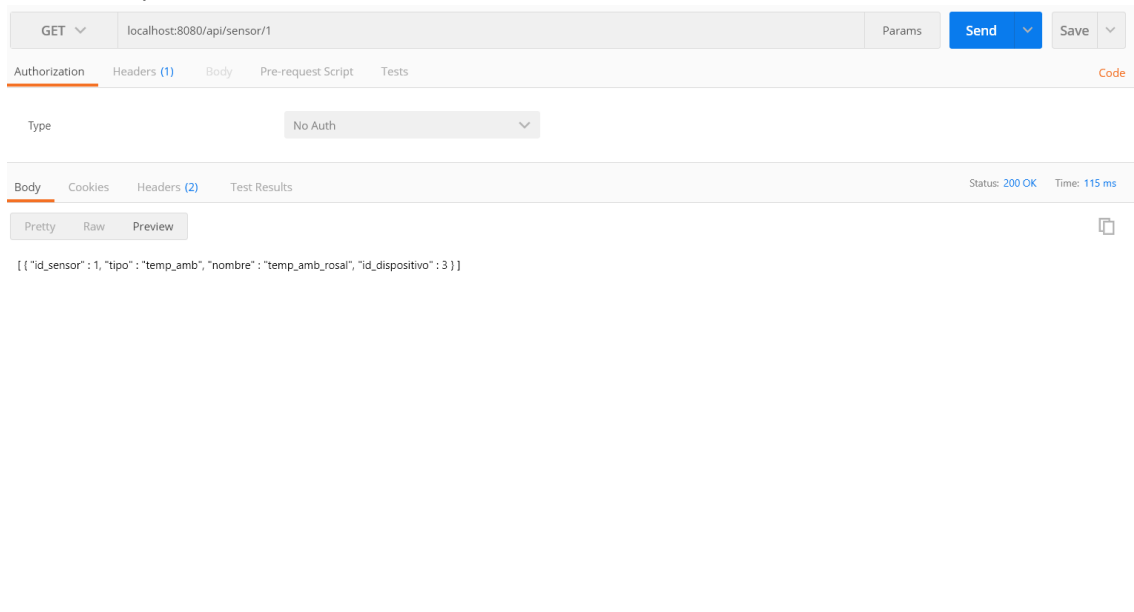
-Url: localhost:8080/api/sensor /:id\_sensor → donde :id\_sensor es la id del sensor del cual queremos hacer la consulta

### -Metodo:

```
private void get_sensor(RoutingContext routingContext) {
    mySQLPool.query("SELECT * FROM dadatabase.sensor WHERE id_sensor = "
        + routingContext.request().getParam("id_sensor"), res -> {
        if (res.succeeded()) {
            RowSet<Row> resultSet = res.result();
            System.out.println("Consulta satisfactoria");
            JSONArray result = new JSONArray();
            for (Row row : resultSet) {
                result.add(JSONObject.mapFrom(new sensor(row.getInteger("id_sensor"),
                    row.getString("tipo"), row.getString("nombre"), row.getInteger("id_dispositivo"))));
            }
            routingContext.response().setStatusCode(200).putHeader("content-type", "application/json")
                .end(result.encodePretty());
        } else {
            System.out.println("Consulta fallida");
            routingContext.response().setStatusCode(401).putHeader("content-type", "application/json")
                .end(JSONObject.mapFrom(res.cause()).encodePretty());
        }
    });
}
```

En el cual se hace una query con un SELECT donde se le pasara la id del sensor, y, si la consulta es satisfactoria, se nos proporcionará un objetos JSONArray con todos los Json que contengan la información de la BBDD. En caso de que sea fallida, se mostrará un mensaje de error.

### -Captura Postman:



## Metodo Put:

-Url: localhost:8080/api/sensor

### -Metodo:

```
//Metodo Put para sensor
private void put_sensor(RoutingContext routingContext) {
    sensor sensor = Json.decodeValue(routingContext.getBodyAsString(), sensor.class);
    mySQLPool.preparedQuery(
        "INSERT INTO daddatabase.sensor (tipo, nombre, id_dispositivo) VALUES (?, ?, ?)",
        Tuple.of(sensor.getTipo(), sensor.getNombre(), sensor.getId_dispositivo()),
        handler -> {
            if (handler.succeeded()) {
                System.out.println("Añadida correctamente");
                System.out.println(handler.result().rowCount());

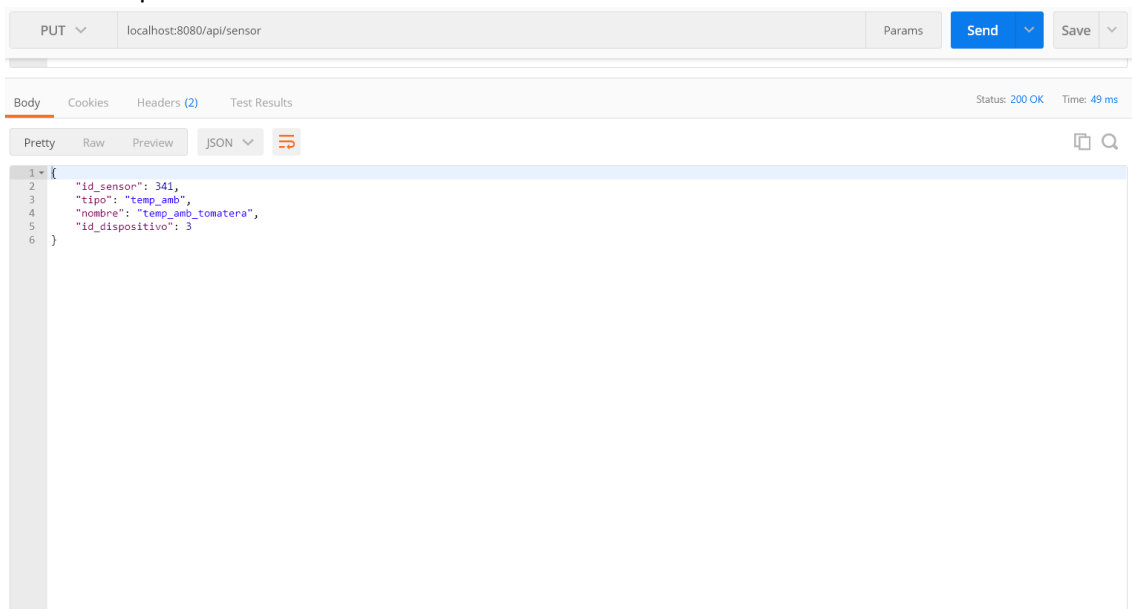
                long id = handler.result().property(MySQLClient.LAST_INSERTED_ID);
                sensor.setId_sensor((int) id);

                routingContext.response().setStatusCode(200).putHeader("content-type", "application/json")
                    .end(JsonObject.mapFrom(sensor).encodePretty());
            } else {
                System.out.println("Algo salió mal");
                System.out.println(handler.cause().toString());
                routingContext.response().setStatusCode(401).putHeader("content-type", "application/json")
                    .end(JsonObject.mapFrom(handler.cause()).encodePretty());
            }
        });
}
```

En el cual se hace una query con un INSERT donde se le pasara los valores de los parámetros que se piden en forma de objeto Json, y, si la prueba es satisfactoria, se añadirá a la información de la BBDD. En caso de que sea fallida, se mostrará un mensaje de error.

```
{
    "tipo": "temp_amb",
    "nombre": "temp_amb_tomatera",
    "id_dispositivo": 3
}
```

### -Captura Postman:



## Entidad dispositivo:

### Metodo Get →

-Url: localhost:8080/api/dispositivo/:id\_dispositivo → donde :id\_dispositivo es la id del dispositivo del cual queremos hacer la consulta

### -Metodo:

```
private void get_dispositivo(RoutingContext routingContext) {  
    MySQLPool.query("SELECT * FROM database.dispositivo WHERE id_dispositivo = "  
    + routingContext.request().getParam("id_dispositivo"), res -> {  
        if (res.succeeded()) {  
            RowSet<Row> resultSet = res.result();  
            System.out.println("Consulta satisfactoria");  
            JSONArray result = new JSONArray();  
            for (Row row : resultSet) {  
                result.add(JSONObject.mapFrom(new dispositivo(row.getInteger("id_dispositivo"),  
                    row.getString("ip"), row.getString("nombre"), row.getInteger("id_planta"), row.getLong("tiempoinicial"))));  
            }  
            routingContext.response().setStatusCode(200).putHeader("content-type", "application/json")  
                .end(result.encodePrettily());  
        } else {  
            System.out.println("Consulta fallida");  
            routingContext.response().setStatusCode(401).putHeader("content-type", "application/json")  
                .end(JSONObject.mapFrom(res.cause()).encodePrettily());  
        }  
    }  
};
```

En el cual se hace una query con un SELECT donde se le pasara la id del dispositivo, y, si la consulta es satisfactoria, se nos proporcionará un objetos JSONArray con todos los Json que contengan la información de la BBDD. En caso de que sea fallida, se mostrará un mensaje de error.

### -Captura Postman:

The screenshot shows the Postman interface for a GET request to `localhost:8080/api/dispositivo/1`. The request is successful with a status of 200 OK and a response time of 238 ms. The response body is displayed in the 'Body' tab, showing a JSON array with one object representing a device.

```
[{"id_dispositivo": 1, "ip": "192.168.0.108", "nombre": "CruzVerde_Aloevera", "id_planta": 1, "tiempoinicial": 123456789}]
```

## Metodo Put:

-Url: localhost:8080/api/dispositivo

### -Metodo:

```
private void put_dispositivo(RoutingContext routingContext) {
    dispositivo dispositivo = Json.decodeValue(routingContext.getBodyAsString(), dispositivo.class);
    mySQLPool.preparedQuery(
        "INSERT INTO daddatabase.dispositivo (ip, nombre, id_planta, tiempoInicial) VALUES (?, ?, ?, ?)",
        Tuple.of(dispositivo.getIp(), dispositivo.getNombre(), dispositivo.getId_planta(), dispositivo.getTiempoInicial()),
        handler -> {
            if (handler.succeeded()) {
                System.out.println("Añadida correctamente");
                System.out.println(handler.result().rowCount());

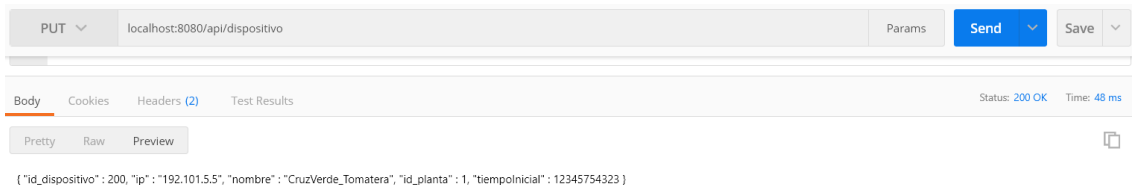
                long id = handler.result().property(MySQLClient.LAST_INSERTED_ID);
                dispositivo.setId_dispositivo((int) id);

                routingContext.response().setStatusCode(200).putHeader("content-type", "application/json")
                    .end(JsonObject.mapFrom(dispositivo).encodePretty());
            } else {
                System.out.println("Algo salió mal");
                System.out.println(handler.cause().toString());
                routingContext.response().setStatusCode(401).putHeader("content-type", "application/json")
                    .end(JsonObject.mapFrom(handler.cause()).encodePretty());
            }
        });
}
```

En el cual se hace una query con un INSERT donde se le pasara los valores de los parámetros que se piden en forma de objeto Json, y, si la prueba es satisfactoria, se añadirá a la información de la BBDD. En caso de que sea fallida, se mostrará un mensaje de error.

```
{
    "ip" : "192.101.5.5",
    "nombre" : "CruzVerde_Tomatera",
    "id_planta" : 4,
    "tiempoInicial" : 12345754323
}
```

### -Captura Postman:



## Entidad planta:

### Metodo Get →

-Url: localhost:8080/api/planta/:id\_planta → donde :id\_planta es la id de la planta de la cual queremos hacer la consulta

### -Metodo:

```
private void get_planta(RoutingContext routingContext) {  
    MySQLPool.query("SELECT * FROM daddatabase.planta WHERE id_planta = "  
        + routingContext.request().getParam("id_planta"), res -> {  
        if (res.succeeded()) {  
            RowSet<Row> resultSet = res.result();  
            System.out.println("Consulta satisfactoria");  
            JSONArray result = new JSONArray();  
            for (Row row : resultSet) {  
                result.add(JSONObject.mapFrom(new planta(row.getInteger("id_planta"),  
                    row.getString("nombre_planta"), row.getFloat("temp_amb_planta"), row.getFloat("humed_tierra_planta"),  
                    row.getFloat("humed_amb_planta"))));  
            }  
            routingContext.response().setStatusCode(200).putHeader("content-type", "application/json")  
                .end(result.encodePrettily());  
        } else {  
            System.out.println("Consulta fallida");  
            routingContext.response().setStatusCode(401).putHeader("content-type", "application/json")  
                .end((JSONObject.mapFrom(res.cause()).encodePrettily()));  
        }  
    });  
}
```

En el cual se hace una query con un SELECT donde se le pasara la id del dispositivo, y, si la consulta es satisfactoria, se nos proporcionará un objetos JSONArray con todos los Json que contengan la información de la BBDD. En caso de que sea fallida, se mostrará un mensaje de error.

### -Captura Postman:

The screenshot shows the Postman interface for a GET request to `localhost:8080/api/planta/1`. The request is successful with a status of 200 OK and a response time of 142 ms. The response body is a JSON array containing one object with the following data:

```
[{"id_planta": 1, "nombre_planta": "Aloe vera", "temp_amb_planta": 22.0, "humed_tierra_planta": 25.0, "humed_amb_planta": 5.0}]
```

## Metodo Put:

-Url: localhost:8080/api/planta

### -Metodo:

```
private void put_planta(RoutingContext routingContext) {
    planta planta = Json.decodeValue(routingContext.getBodyAsString(), planta.class);
    MySQLPool.preparedQuery(
        "INSERT INTO daddatabase.planta (nombre_planta, temp_amb_planta, humed_tierra_planta, humed_amb_planta) VALUES (?, ?, ?, ?)"]
    Tuple.of(planta.getNombre_planta(), planta.getTemp_amb_planta(), planta.getHumed_tierra_planta(), planta.getHumed_amb_planta()),
    handler -> {
        if (handler.succeeded()) {
            System.out.println("Añadida correctamente");
            System.out.println(handler.result().rowCount());

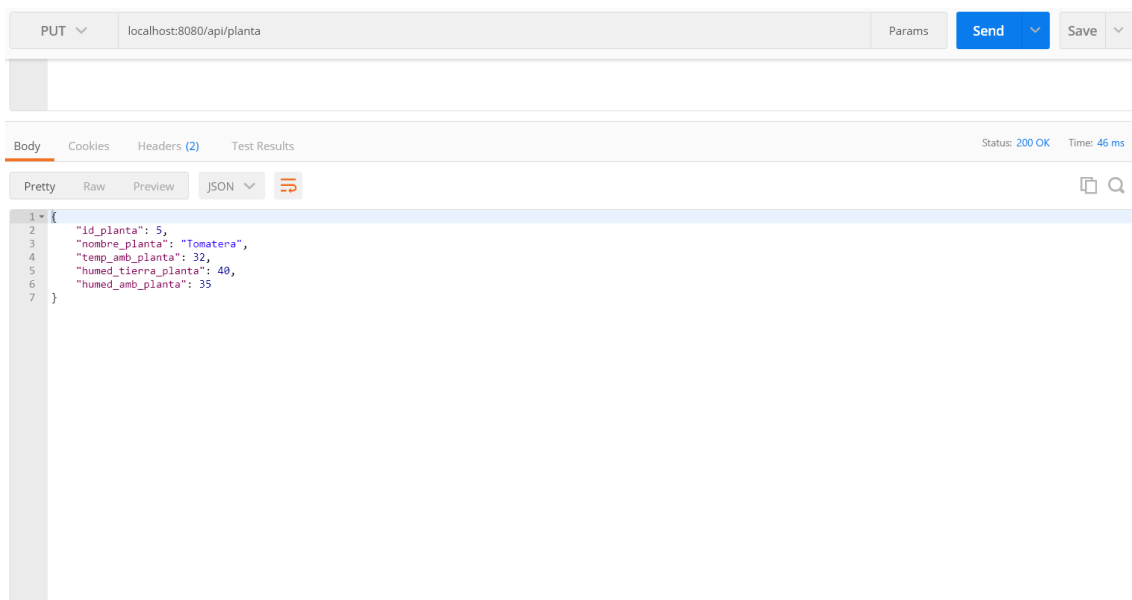
            long id = handler.result().property(MySQLClient.LAST_INSERTED_ID);
            planta.setId_planta((int) id);

            routingContext.response().setStatusCode(200).putHeader("content-type", "application/json")
                .end(JsonObject.mapFrom(planta).encodePretty());
        } else {
            System.out.println("Algo salió mal");
            System.out.println(handler.cause().toString());
            routingContext.response().setStatusCode(401).putHeader("content-type", "application/json")
                .end(JsonObject.mapFrom(handler.cause()).encodePretty());
        }
    });
}
```

En el cual se hace una query con un INSERT donde se le pasara los valores de los parámetros que se piden en forma de objeto Json, y, si la prueba es satisfactoria, se añadirá a la información de la BBDD. En caso de que sea fallida, se mostrará un mensaje de error.

```
{
    "nombre_planta": "Tomatera",
    "temp_amb_planta": 32,
    "humed_tierra_planta": 40,
    "humed_amb_planta": 35
}
```

### -Captura Postman:





Las entidades `actuador` y `actuador_valor`, no tienen ninguna utilidad, están diseñados sus métodos pero no serán necesarios.