ORACLE

Growing in the Cloud - My Journey to Create Hot Sauce Using IoT, Messaging, Micronaut and Autonomous DB to Automate and Monitor Seedling Growth

Todd Sharp
Developer Advocate - Cloud & Cloud DB
todd.sharp@oracle.com
@recursivecodes
Q2 - 2021

ORACLE

In this session, I'll show you how I used microcontrollers, sensors, and the cloud to create the worlds greatest* hot sauce.

*Based on my opinion


THE MOTIVATION

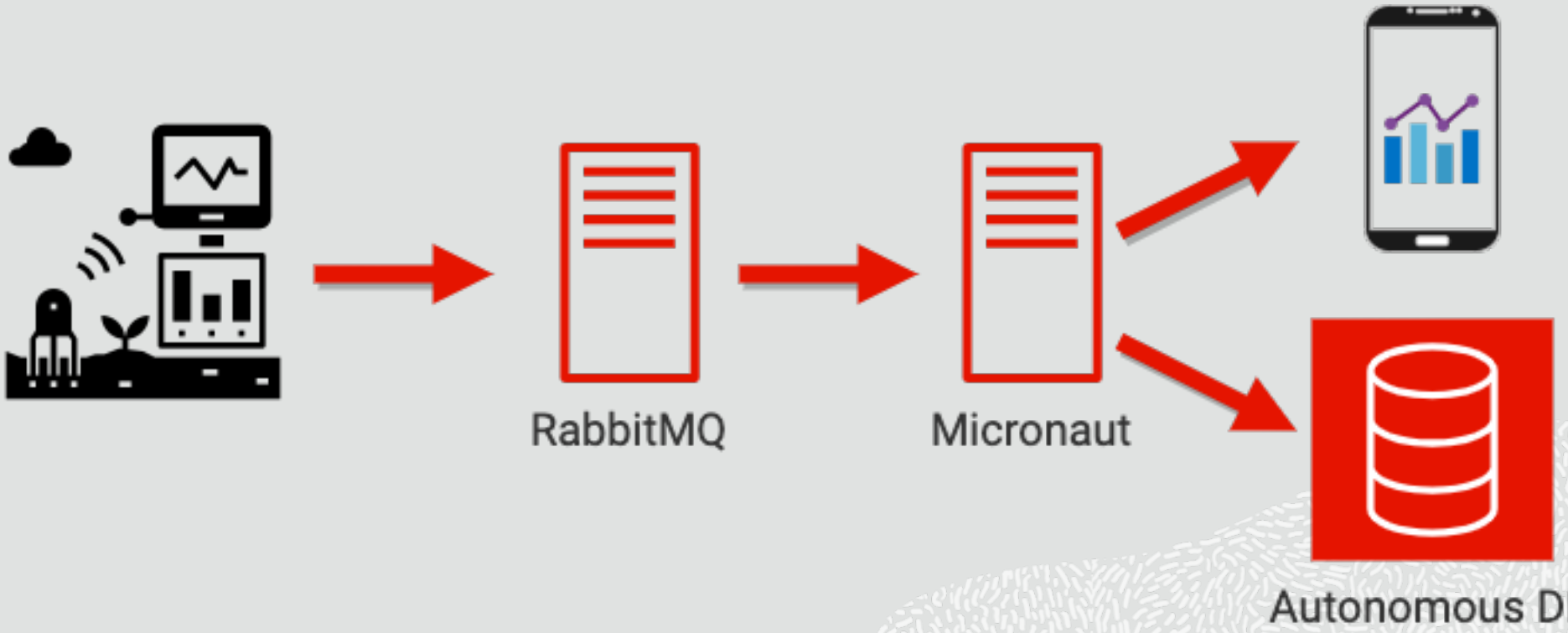
The Objective

Time Period	Air Temp	Soil Temp	Moisture	Humidity	Light
Day	65°-80°F 18°-26°C	80°F 26°C	70-80%	50-70%	>1600 lux
Night	60°-70°F 15°-21°C	70°F 21°C	70-80%	50-70%	>1600 lux

ORACLE

The Hardware Build

The Architecture



ORACLE

The Microcontroller Code

```
#define WATER_SENSOR A0
moisture = analogRead(WATER_SENSOR);
```

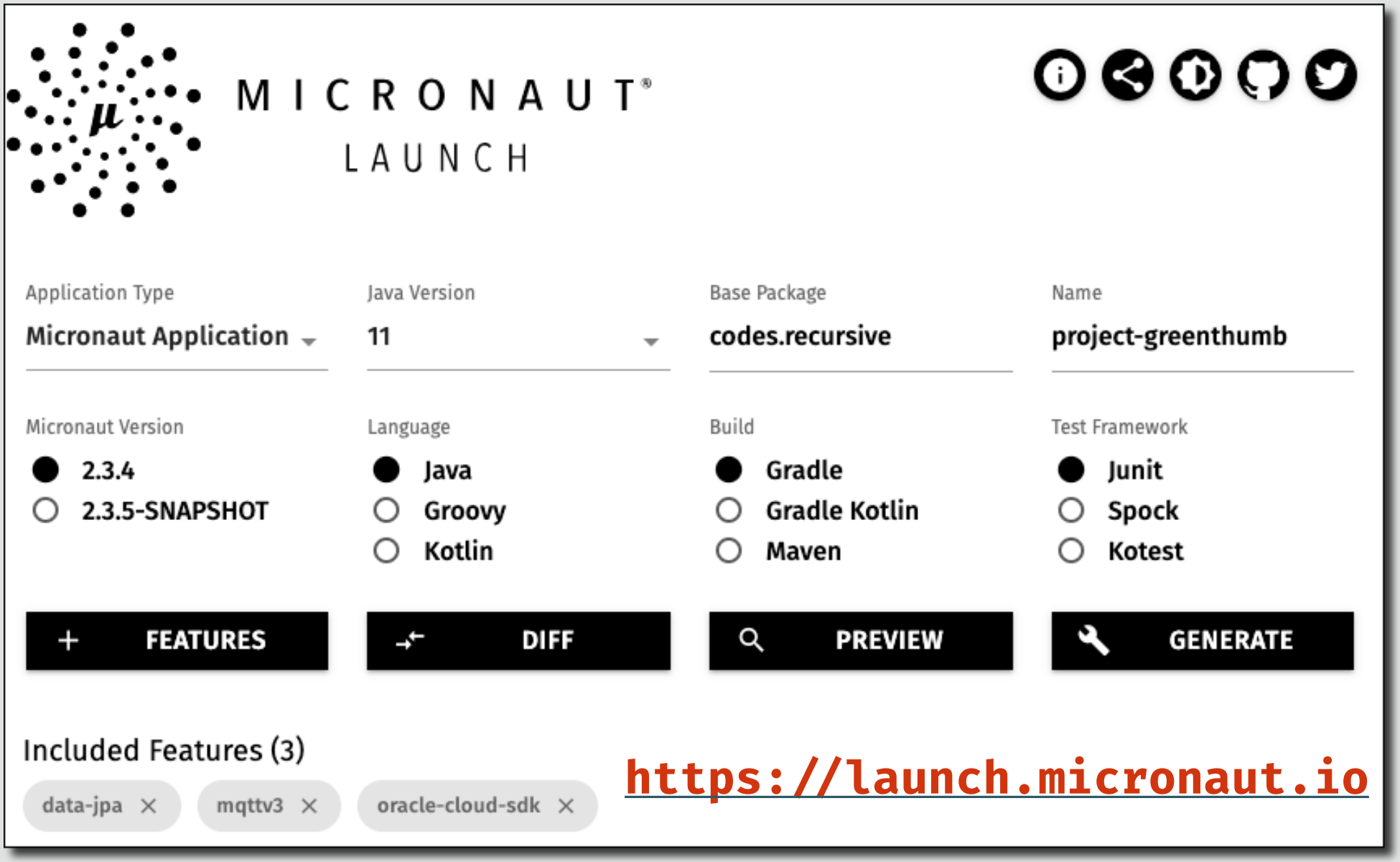
```
#include <dht.h>
#define DHT11_PIN 10
DHT.read11(DHT11_PIN);
int tempC = DHT.temperature
int humidity = DHT.humidity
```

```
// Include the library
#include <ArduinoJson.h>

// Create a JSON Document & String to Hold Serialized Result
StaticJsonDocument<150> doc;
char readingsJson[256];

// Set Document Values & Serialize
doc["outletState"] = relayState;
doc["airTemp"] = cToF(DHT.temperature);
doc["soilTemp"] = probeTempF;
doc["humidity"] = DHT.humidity;
doc["moisture"] = moisturePct;
doc["light"] = lux;

serializeJson(doc, readingsJson);
```

ORACLE

The Persistence & Visualization App

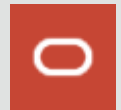
```
CREATE TABLE GREENTHUMB_READINGS(
    "ID" NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
    "READING" CLOB NOT NULL,
    "CREATED_ON" TIMESTAMP(6) NOT NULL,
    CONSTRAINT ensure_json CHECK (READING IS JSON),
    CONSTRAINT "GREENTHUMB_READINGS_PK" PRIMARY KEY ("ID")
);
```


MICRONAUT®
LAUNCH
Application Type
Micronaut Application
Java Version
11
Base Package
codes.recursive
Name
project-greenthumb
Micronaut Version
2.3.4
2.3.5-SNAPSHOT
Language
Java
Groovy
Kotlin
Build
Gradle
Gradle Kotlin
Maven
Test Framework
Junit
Spock
Kotest
FEATURES
DIFF
PREVIEW
GENERATE
Included Features (3)
data-jpa
mqttv3
oracle-cloud-sdk
https://launch.micronaut.io

```yaml
# application-local.yml

mqtt:
  client:
    userName: todd
    password: passwerd_123
    serverUri: tcp://myrabbitmq.mydomain.com
    clientId: greenthumb-local
    connectionTimeout: 30s
    automaticReconnect: true
```

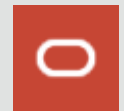
```java
@MqttSubscriber
public class GreenThumbConsumer {
    @Topic("greenthumb/readings")
    public void receive(Map<String, Object> data) {
        Reading reading = new Reading(data);
    }
}
```

```
1 14:47:57.789 [MQTT Call: greenthumb] TRACE i.m.m.i.AbstractMqttSubscriberAdvice - Qos = 0, MessageId = 0,
  Payload = {"outletState":1,"airTemp":71.96,"soilTemp":79.3625,"humidity":37,"moisture":79,"light":959}
2 14:48:08.569 [MQTT Call: greenthumb] TRACE i.m.m.i.AbstractMqttSubscriberAdvice - Received the following
  message from greenthumb/readings
3 14:48:08.569 [MQTT Call: greenthumb] TRACE i.m.m.i.AbstractMqttSubscriberAdvice - Qos = 0, MessageId = 0,
  Payload = {"outletState":1,"airTemp":71.96,"soilTemp":79.3625,"humidity":31,"moisture":79,"light":956}
4 14:48:19.362 [MQTT Call: greenthumb] TRACE i.m.m.i.AbstractMqttSubscriberAdvice - Received the following
  message from greenthumb/readings
5 14:48:19.362 [MQTT Call: greenthumb] TRACE i.m.m.i.AbstractMqttSubscriberAdvice - Qos = 0, MessageId = 0,
  Payload = {"outletState":1,"airTemp":71.96,"soilTemp":79.3625,"humidity":27,"moisture":79,"light":957}
```

```java
@Entity
@Table(name = "greenthumb_readings")
public class Reading {
    @Id
    @GeneratedValue(strategy= GenerationType.IDENTITY)
    private Long id;
    private String reading;
    private Date createdOn;
}
```


```java
@Repository
public interface ReadingRepository extends PageableRepository<Reading, Long> {
    CompletableFuture<Reading> saveAsync(@NonNull Reading reading);
}
```


select
id,
gr.reading.airTemp,
created_on
from greenthumb_readings gr

```
select
    to_char(gr.created_on, 'HH24') as hour,
    round(avg(gr.reading.airTemp), 2) as airTemp
from greenthumb_readings gr
group by to_char(gr.created_on, 'HH24')
order by to_char(created_on, 'HH24');
```

```java
@ServerWebSocket("/data/{topic}")
public class GreenThumbWebSocket {
    private final WebSocketBroadcaster broadcaster;
    public GreenThumbWebSocket(WebSocketBroadcaster broadcaster) {
        this.broadcaster = broadcaster;
    }

    @OnOpen
    public void onOpen(String topic, WebSocketSession session) {
        broadcaster.broadcastSync("Joined channel", isValid(topic, session));
    }

    @OnMessage
    public void onMessage(String topic, String message, WebSocketSession session) {
        broadcaster.broadcastSync(message, isValid(topic, session));
    }

    @OnClose
    public void onClose(String topic, WebSocketSession session) {
        broadcaster.broadcastSync("Disconnected", isValid(topic, session));
    }

    private Predicate<WebSocketSession> isValid(String topic, WebSocketSession session) {
        return s -> s != session &&
                topic.equalsIgnoreCase(s.getUriVariables().get("topic", String.class, null));
    }
}
```

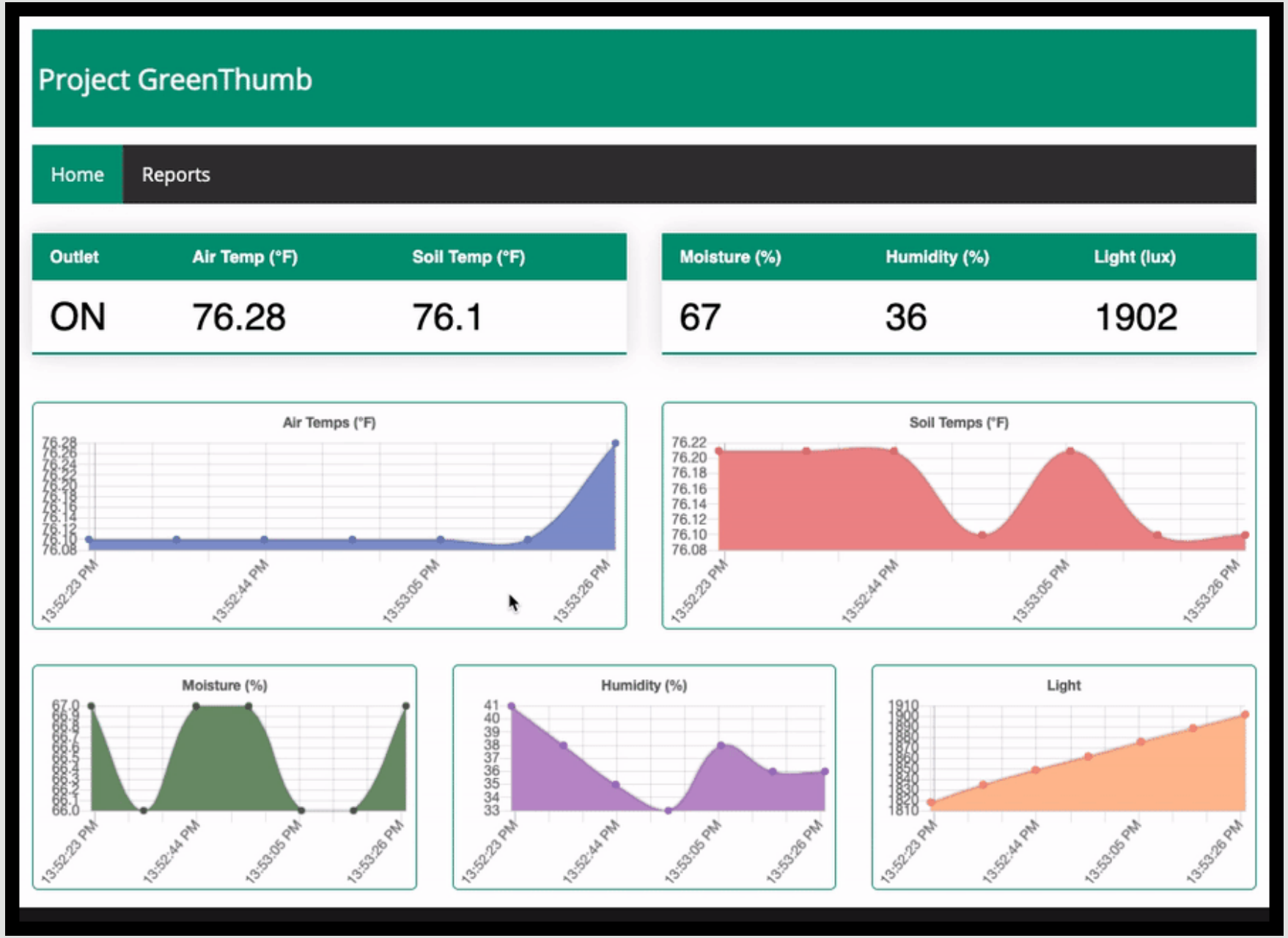
```
public void receive(Map<String, Object> data) throws JsonProcessingException {
        Reading reading = new Reading(data);
        // persist to the DB
        readingRepository.saveAsync(reading);
        // broadcast to websocket client
        broadcaster.broadcastAsync(data, MediaType.APPLICATION_JSON_TYPE);
        // push notification
        int soilMoisture = (int) reading.getReadingAsMap().get("moisture");
        if( (System.currentTimeMillis() - lastAlert > interval)
           && soilMoisture < moistureThreshold) {
            PushNotificationResponse response = pushoverClient.pushMessage(
                    apiKey,
                    userKey,
                    "Soil Moisture Alert! Current moisture: " + soilMoisture,
                    "http://greenthumb.toddrsharp.com:8080/page"
            ).blockingFirst();
            lastAlert = System.currentTimeMillis();
        }
}
```

ORACLE

Simple Views

```
@Get()
ModelAndView home() {
    return new ModelAndView(
      "home",
      CollectionUtils.mapOf("currentView", "home")
    );
}
```

```
const connect = () => {
    console.log('Connecting to WebSocket...')
    const ws = new WebSocket("ws://" + location.hostname + ":" + location.port + "/data/greenthumb");
    ws.onopen = (msg) => {
        console.log('Connected!')
    };
    ws.onmessage = (msg) => {
        const reading = new Reading(JSON.parse(msg.data));
        soilTempReadings.push({y: reading.soilTemp, x: reading.readAt});
        // keep the latest `maxPoints`
        if( soilTempReadings.length > maxPoints ) soilTempReadings.shift();
        if (chartsInit)  soilTempChart.update();

        // update the table that displays the latest values
        document.querySelector('#currentOutletState').innerHTML = reading.outletState;
        // update other readings...
        document.querySelector('#currentLight').innerHTML = reading.light;
    };
    ws.onclose = (e) => {
        console.log('Socket is closed. Reconnect will be attempted in 1 second.', e.reason);
        setTimeout(function() {
            connect();
        }, 1000);
    };
    ws.onerror = function(err) {
        console.error('Socket encountered error: ', err.message, 'Closing socket');
        ws.close();
    };
};
```


Project GreenThumb
Home
Reports
Outlet
Air Temp (°F)
Soil Temp (°F)
ON
76.28
76.1
Moisture (%)
Humidity (%)
Light (lux)
67
36
1902
Air Temps (°F)
Soil Temps (°F)
Moisture (%)
Humidity (%)
Light
13:52:23 PM
13:52:44 PM
13:53:05 PM
13:53:26 PM

Average Readings Per Hour of Day (Today)

Year	Hour	Avg Air Temp (°F)	Avg Soil Temp (°F)	Avg Humidity (%)	Avg Moisture (%)	Avg Light (lux)
2021	0	67.8	70.65	46.11	72.12	0.0
2021	1	66.98	70.33	46.64	70.79	0.0
2021	2	66.3	68.92	46.94	69.67	0.0
2021	3	65.75	71.47	46.84	67.75	0.0

Average Readings Per Hour of Day (All Time)

Year	Hour	Avg Air Temp (°F)	Avg Soil Temp (°F)	Avg Humidity (%)	Avg Moisture (%)	Avg Light (lux)
2021	0	65.41	70.49	53.91	74.7	0.0
2021	1	64.88	69.78	54.41	74.15	0.09
2021	2	64.67	70.1	50.16	73.36	3.89
2021	3	64.31	70.12	46.15	72.27	0.0

Average Readings - Day vs. Night (All Time)

Year	Period	Avg Air Temp (°F)	Avg Soil Temp (°F)	Avg Humidity (%)	Avg Moisture (%)	Avg Light (lux)
2021	Night	65.17	70.61	50.88	73.45	6.18
2021	Day	70.57	78.04	47.42	77.01	4168.03

The Results

Time Period	Air Temp	Soil Temp	Moisture	Humidity	Light
Day Objective	65°-80°F	80°F	70-80%	50-70%	>1600 lux
Day Result	70.57°F	78.04°F	77%	47%	4168 lux
Night Objective	60°-70°F	70°F	70-80%	50-70%	>1600 lux
Night Result	65.17°F	70.61°F	73%	51%	6.18 lux

Links

- https://blogs.oracle.com/developers/project-greenthumb-part-1-automating-monitoring-seedling-growth-with-microcontrollers-the-cloud
- https://blogs.oracle.com/developers/project-greenthumb-part-2-the-data-collection
- https://blogs.oracle.com/developers/project-greenthumb-part-3-consuming-and-persisting-the-sensor-data-in-the-cloud
- https://blogs.oracle.com/developers/project-greenthumb-part-4-reporting-queries-and-websockets
- https://blogs.oracle.com/developers/project-greenthumb-part-5-the-front-end,-build-pipeline,-push-notifications-and-overall-progress
- https://github.com/recursivecodes/project-greenthumb
- https://github.com/recursivecodes/project-greenthumb-microcontroller

More Links

- https://arduinojson.org
-

Parts

- NodeMCU
- Relay (outlet)
- DHT11 (temp + humidity)
- BH1750 (light)
- OLED Display
- DS18B20 (Probe Thermometer)
- Soil Moisture
- Heat Mat

Socials & Contact

- https://blogs.oracle.com/author/todd-sharp
- https://recursive.codes
- https://twitter.com/recursivecodes
- https://www.linkedin.com/in/toddrsharp/
- https://github.com/recursivecodes
- todd.sharp@oracle.com