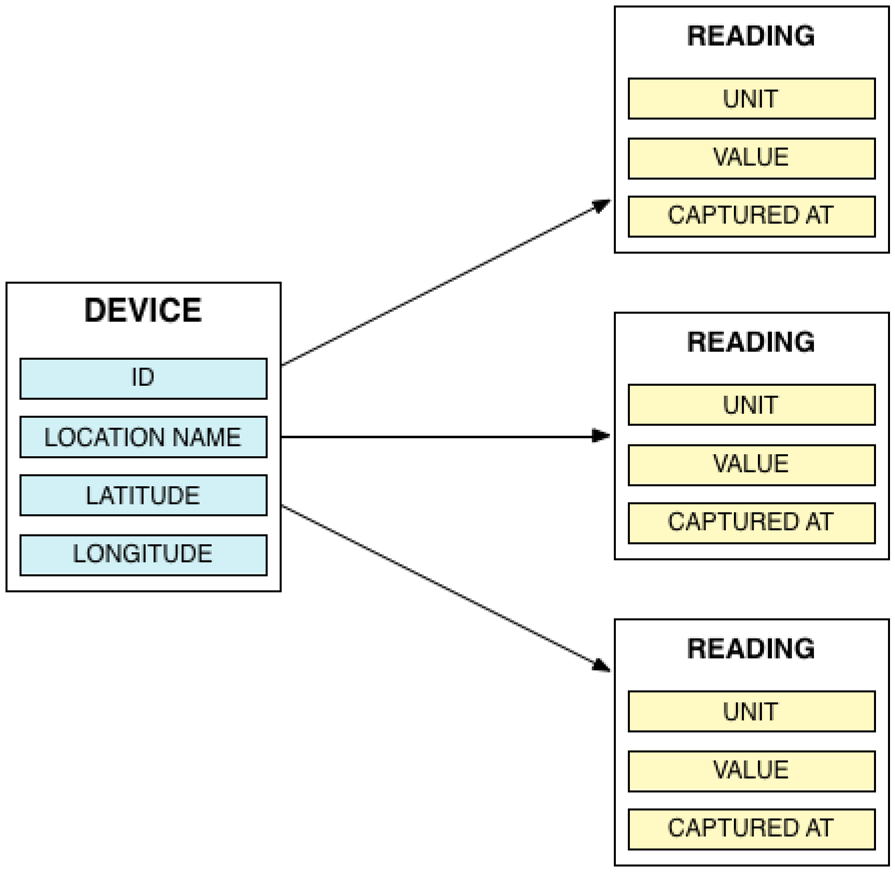
Radiation Sensor

Overview

We'll store device details and store sensor readings over time.



1. Define a Device

We need to create a form that allows you to add new devices. Think of this form as a channel or feed. It is basically a template that defines the data that will be accepted. Each device is defined by:

- Device ID
- Location Name
- Latitude
- Longitude

```
POST /api/v2/forms
```

```
{
  "name": "Radiation Sensor",
  "description": "Defines a single radiation sensor device",
  "type": "report",
  "groups":[
    {
      "label": "Sensor",
      "attributes": [
        {
          "key": "device_id",
          "label": "Device ID",
          "type": "varchar",
          "required": true,
          "input": "text"
        },
        {
          "key": "location_name",
          "label": "Location Name",
          "type": "varchar",
          "required": true,
          "input": "text"
        },
        {
          "key": "latitude",
          "label": "Latitude",
          "type": "decimal",
          "required": true,
          "input": "text"
        },
        {
          "key": "longitude",
          "label": "Longitude",
          "type": "decimal",
          "required": true,
          "input": "text"
        },
      ]
    }
  ]
}
```

2. Define a Sensor Reading

Next we need to define the data that each sensor device will collect. You can think of this as defining the sensor data stream and what data it collects. In this case:

- Unit (cpm's)
- Value
- Capture Date/Time

parent_id references the ID of the form we just created above. The sensor reading form belongs to the device form.

```
POST /api/v2/forms
```

```
{
    "parent_id":1,
    "name": "Sensor Readings",
    "description": "Radiation sensor cpm readings",
    "type": "report",
    "groups":[
        {
            "label": "Reading",
            "attributes": [
                {
                    "key": "unit",
                    "label": "Unit",
                    "type": "varchar",
                    "required": true,
                    "input": "text"
                },
                {
                    "key": "value",
                    "label": "Value",
                    "type": "int",
                    "required": true,
                    "input": "text"
                },
                {
                    "key": "captured_at",
                    "label": "Captured At",
                    "type": "datetime",
                    "required": true,
                    "input": "text"
                }
            ]
        }
    ]
}
```

3. Create a Sensor Device

Lets create a new device! Each of these devices will have an endpoint to receive sensor data.

```
POST /api/v2/posts
```

```
{
   "form_id": 1,
   "locale":"en_US",
   "type": "report",
   "status": "published",
   "title": "Sensor 20823609",
   "values":{
      "device_id": "20823609",
      "latitude": 34.669824,
      "longitude": 138.949453,
      "location_name": "Kitadai-7-2 Ottozawa Okuma"
   }
}
```

4. Create a Sensor Reading

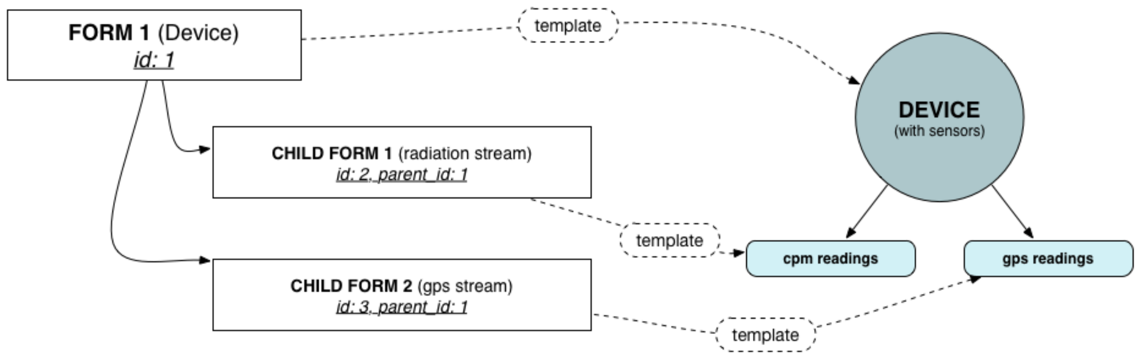
With a device in place, we can start pushing regular updates. If the post:id of the device is 1, the endpoint for sensor data would be /api/v2/posts/1/updates.

NOTE: We can push multiple data streams updates from a device to Ushahidi. form_id below denotes which datastream we're pushing into.

```
POST /api/v2/posts/:id/updates
```

```
{
   "form_id": 2,
   "locale":"en_US",
   "type": "update",
   "status": "published",
   "values":{
      "unit": "cpm",
      "value": 20,
      "captured_at": "2013-03-10T07:43:41Z"
   }
}
```

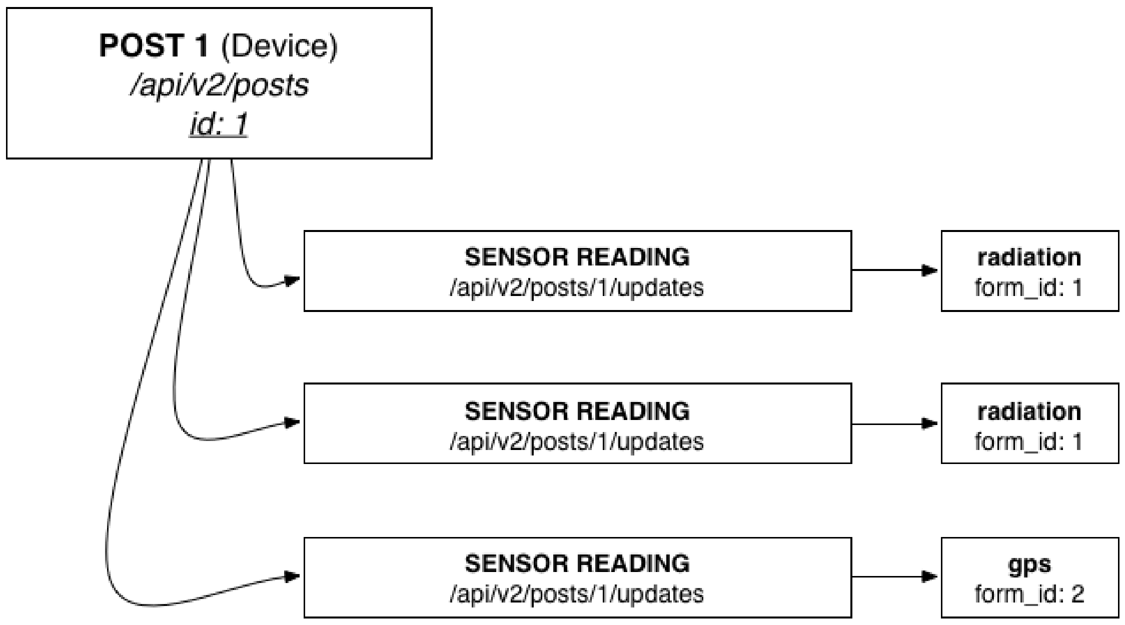
Forms



At the core of Ushahidi is posts and posts are defined by forms. Forms are basically templates. They allow the system to structure the data correctly by defining what is expected along with making sure it is valid. Forms can have sub-forms. The reason for this is illustrated above. You want one form to define what a device is. Next you want another form to define the datastreams generated by this device.

Why not have a single form? Each time you receive a new sensor reading, having a single form with the device information and sensor readings just means you'd have a lot more columns and some like the device information that doesn't change over time being repeated over and over, consuming unnecessary resources.

Posts & Endpoints



- The device we created above is accessible at GET:/api/v2/posts/1, where 1 is the id of the device.
- The endpoint that sensors would transmit readings to is accessible at POST:/api/v2/posts/1/updates
- form_id lets the endpoint know which datastream is posting to. In this case form_id:1 is for the radiation readings and form_id:2 is for lat/lon