CLOUD STORAGE && WEB COMMUNICATION APIs

<u>**CLOUD STORAGE MODELS and COMMUNICATION APIs**</u>

1. WAMP – AutoBahn for IoT
2. **Xively Cloud for IoT**
3. Python Web Application Framework – Django
4. Designing RESTful web API
5. Amazon web services for IoT
6. SkyNet IoT Messaging Platform

Siddaganga Institute of Technology

Xively Cloud for IoT

Xively Cloud for IoT

❖ What is Xively?

- Xively is a commercial **Platform-as-a-Service**
- Xively can be used **for creating solutions** for Internet of Things
- With Xively cloud, IoT developers can focus on
  - ❑ **the front end infrastructure**
  - ❑ **Devices for IoT**
  - ❑ **Management of back end data collection infrastructure**

❖ Xively Platform:

Comprises of:

- **Message bus** for real time message management
- **Data Services** for time series archiving
- **Directory services** for device provisioning and management

Siddaganga Institute of Technology

Xively Cloud for IoT

❖**Advantages of Xively:**

- **Xively** provides an extensive support for various languages and platforms

- **Xively** libraries leverage standards based API over HTTP, Sockets and MQTT for connecting IoT devices to the Xively Cloud

❖**How to use Xively?**

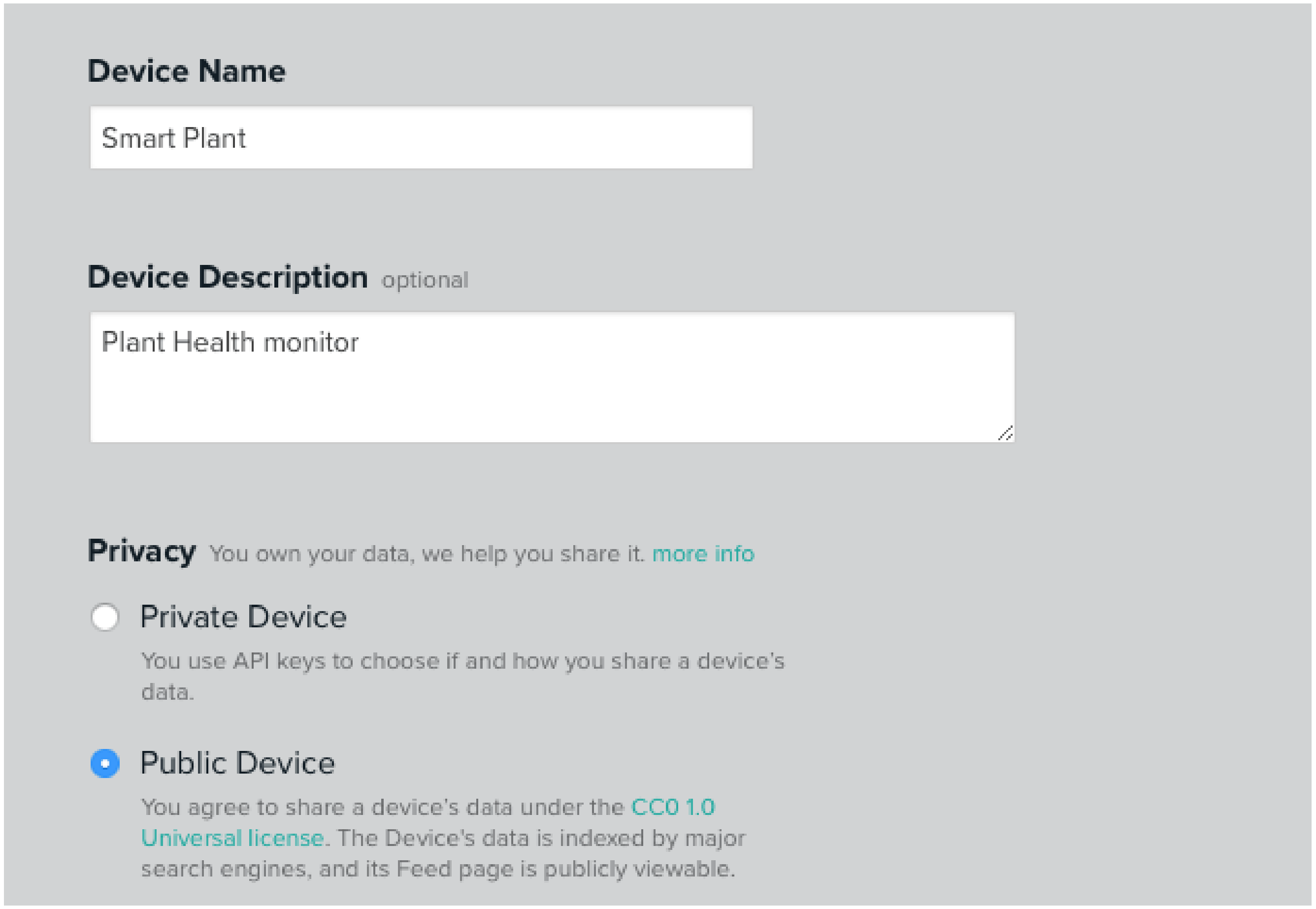
- To start using Xively, one need to register for a developer account
- Then, create development device on Xively (as shown in screen shot – refer pg.5(next slide) of this ppt)
- When device is created, Xively automatically creates a Feed-ID and an API Key to connect to the device

  => Each device has s unique Feed-ID.

"Feed-ID is a **collection of channels** or datastreams defined for a device and the associates meta-data"

"API key s are used to provide different levels of **permissions**. The default API key has **read, update, create** and **delete** permissions"

Xively Cloud : Dashboard


Device Name

Smart Plant

Device Description optional

Plant Health monitor

Privacy You own your data, we help you share it. more info

( ) Private Device

You use API keys to choose if and how you share a device's data.

[✓ Public Device]

You agree to share a device's data under the CC0 1.0 Universal license. The Device's data is indexed by major search engines, and its Feed page is publicly viewable.

Creating a new device

P.S: Xively Cloud has become a Google Cloud now

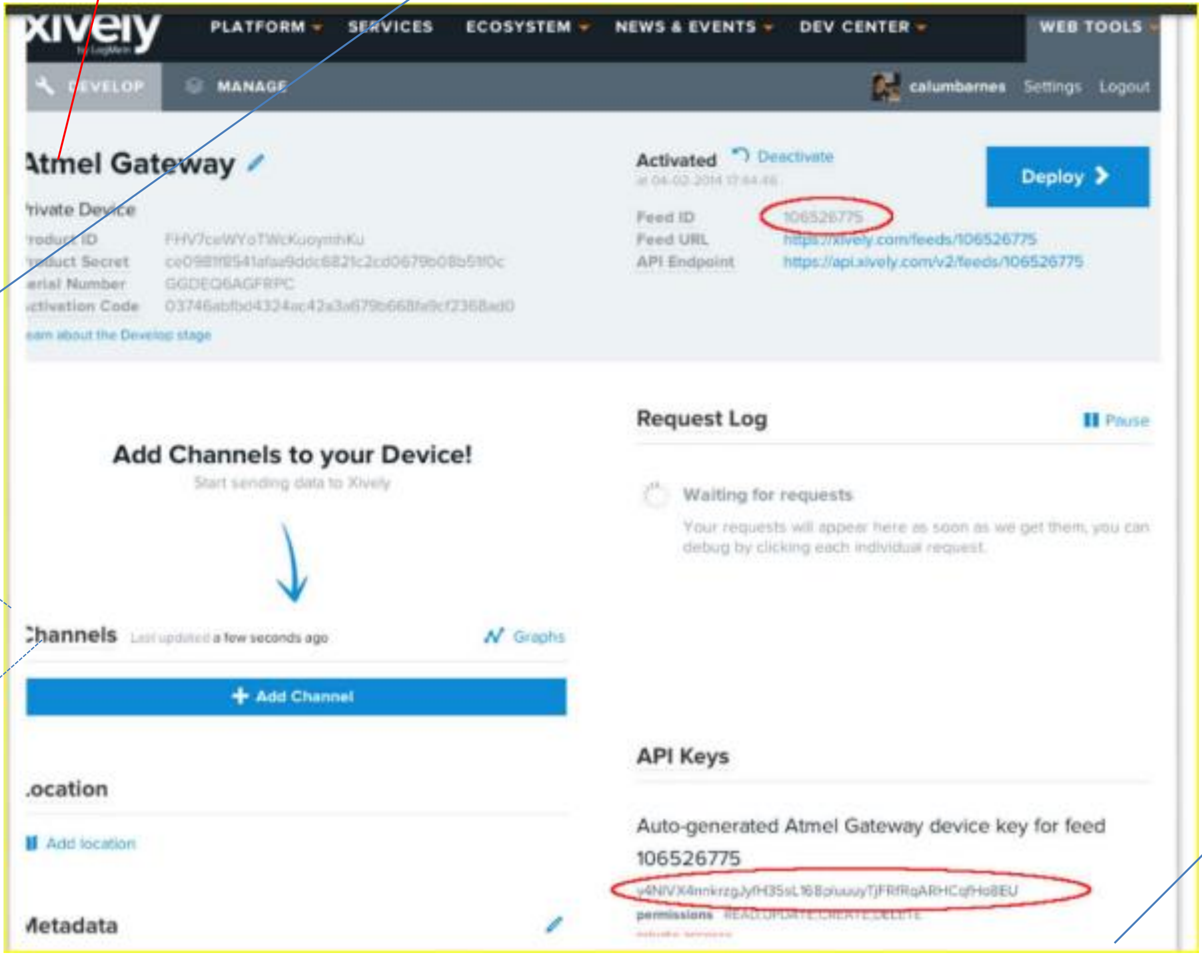
Xively Cloud : Dashboard

IoT device can **send data to a channel** using **Xively APIs**.

For each channel, you can create one or more **triggers**. **Triggers** are used for integration with third party applications

Xively devices can have one or more **channels**

Each channel enables **bi directional** communication between IoT devices and Xively Cloud.



Atmel Gateway

Private Device

Product ID	FHV7ceWYoTWcKuoymhKu
Product Secret	ce0981f2541afaa9ddc6821c2cd0679b08b51f0c
Serial Number	GGDEQ6AGFRPC
Activation Code	03746abfbd4324ac42a3a679b668fa9cf2368ad0

Learn about the Develop stage

Activated Deactivate
at 04-02-2014 17:44:46

Feed ID	106526775
Feed URL	https://xively.com/feeds/106526775
API Endpoint	https://api.xively.com/v2/feeds/106526775

Request Log

Waiting for requests

Your requests will appear here as soon as we get them, you can debug by clicking each individual request.

Add Channels to your Device!
Start sending data to Xively

Channels Last updated a few seconds ago

Location

Metadata

API Keys

Auto-generated Atmel Gateway device key for feed 106526775
y4NIVX4nnkrzgJyfH35sL168pluuuyTJFRfRqARHCqfHo8EU
permissions READ,UPDATE,CREATE,DELETE

New device Details

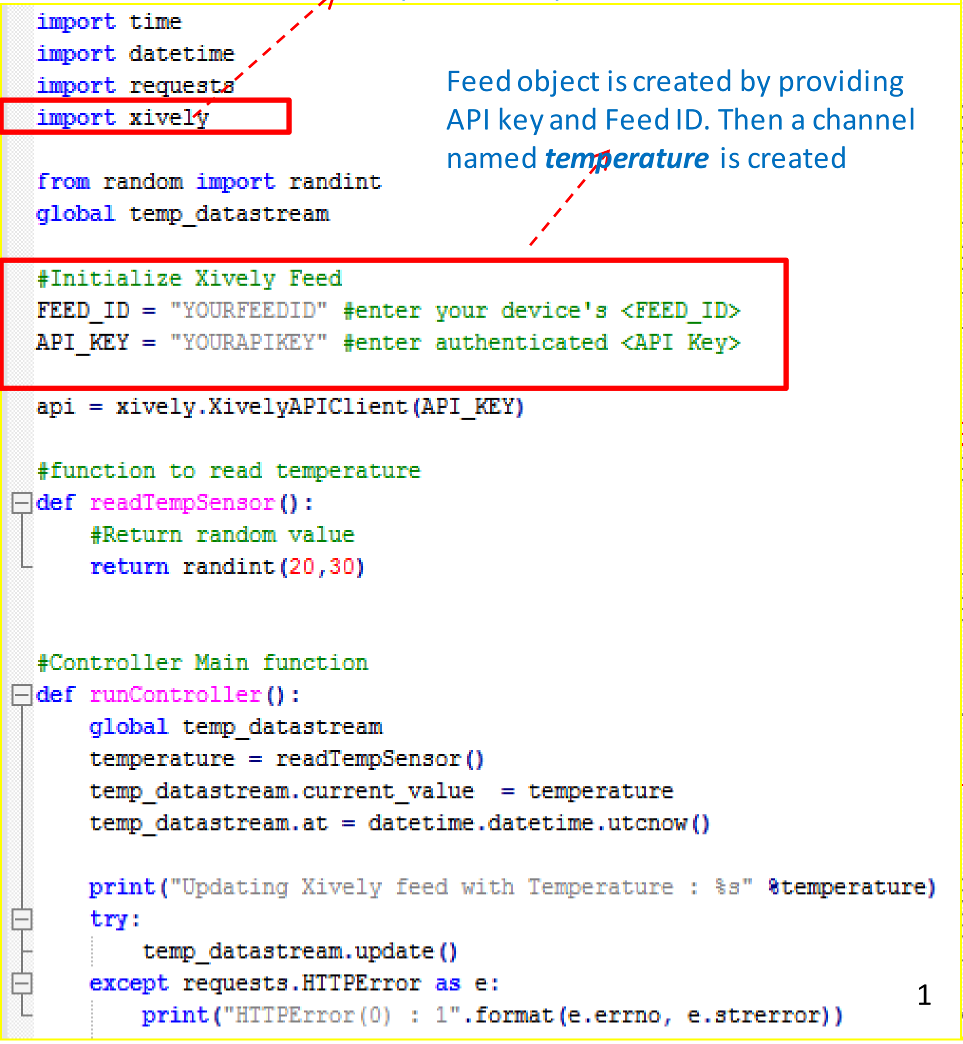
A ***trigger Specification*** includes a channel to which the trigger corresponds, trigger condition and an **HTTP POST URL** to which the request Is sent when trigger fires.

Xively Cloud : Python Program for sending data to a Xively Cloud

Background: temperature monitoring using Raspberry Pi and the measured data is sent to Xively cloud. Raspberry Pi runs a controller program that reads the sensor data(e.g.DHT11) every few seconds and sends data to the cloud.

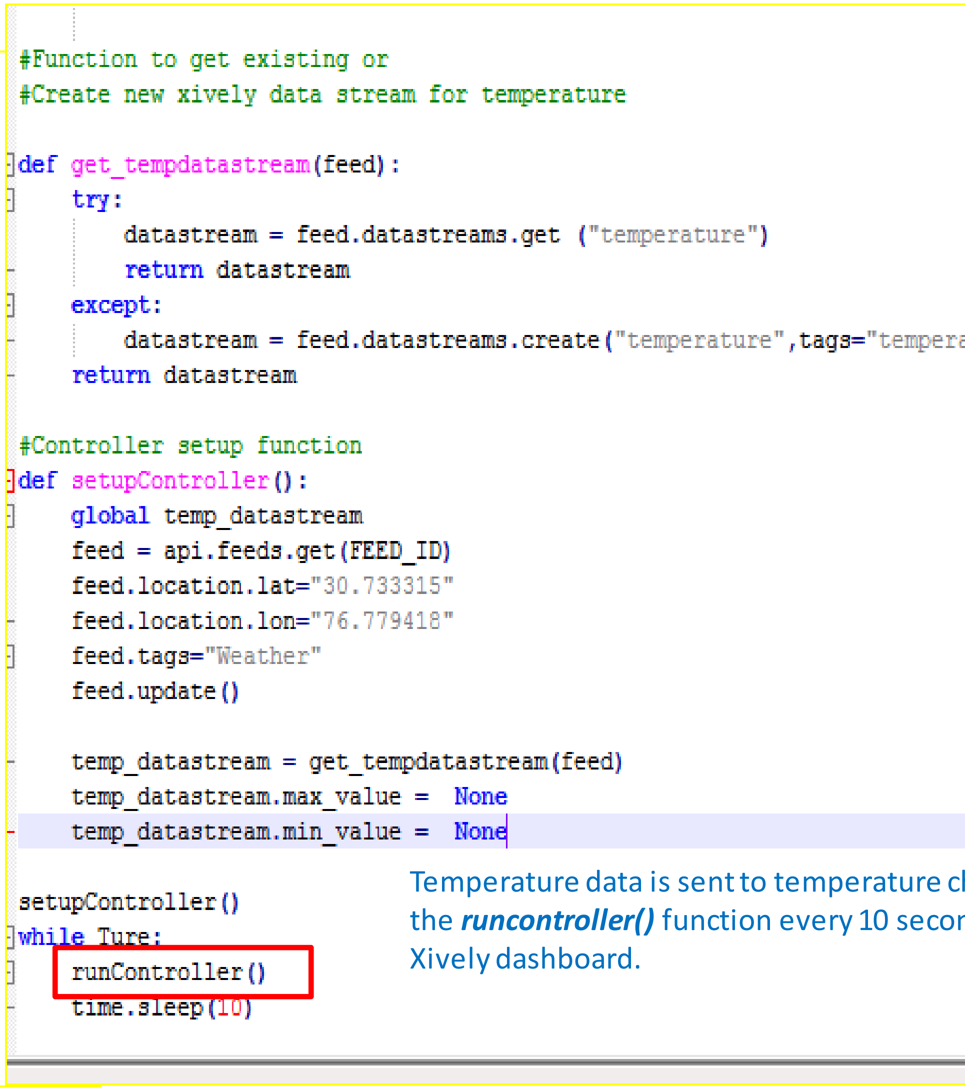
A Python library, needed to execute the code

Feed object is created by providing API key and Feed ID. Then a channel named ***temperature*** is created



```
import time
import datetime
import requests
import xively

from random import randint
global temp_datastream

#Initialize Xively Feed
FEED_ID = "YOURFEEDID" #enter your device's <FEED_ID>
API_KEY = "YOURAPIKEY" #enter authenticated <API Key>

api = xively.XivelyAPIClient(API_KEY)

#function to read temperature
def readTempSensor():
    #Return random value
    return randint(20,30)


#Controller Main function
def runController():
    global temp_datastream
    temperature = readTempSensor()
    temp_datastream.current_value  = temperature
    temp_datastream.at = datetime.datetime.utcnow()

    print("Updating Xively feed with Temperature : %s" %temperature)
    try:
        temp_datastream.update()
    except requests.HTTPError as e:
        print("HTTPError(0) : 1".format(e.errno, e.strerror))
```

1



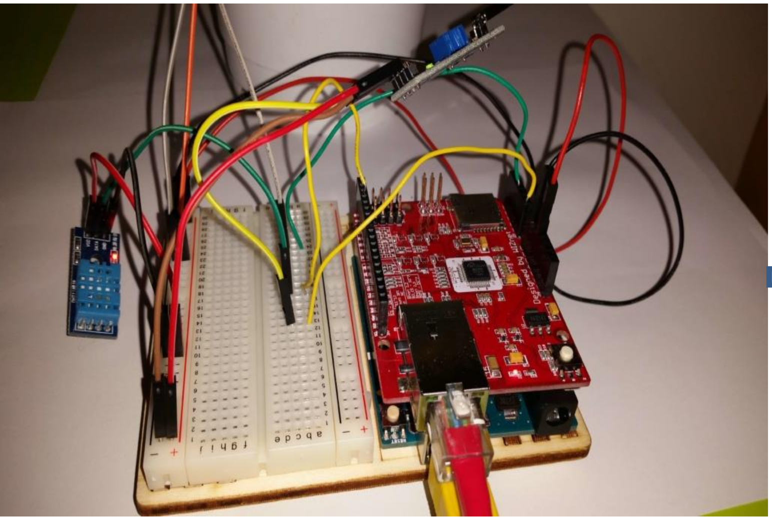
```
#Function to get existing or
#Create new xively data stream for temperature

def get_tempdatastream(feed):
    try:
        datastream = feed.datastreams.get ("temperature")
        return datastream
    except:
        datastream = feed.datastreams.create("temperature",tags="tempera
    return datastream

#Controller setup function
def setupController():
    global temp_datastream
    feed = api.feeds.get(FEED_ID)
    feed.location.lat="30.733315"
    feed.location.lon="76.779418"
    feed.tags="Weather"
    feed.update()

    temp_datastream = get_tempdatastream(feed)
    temp_datastream.max_value =  None
    temp_datastream.min_value =  None

setupController()
while Ture:
    runController()
    time.sleep(10)
```

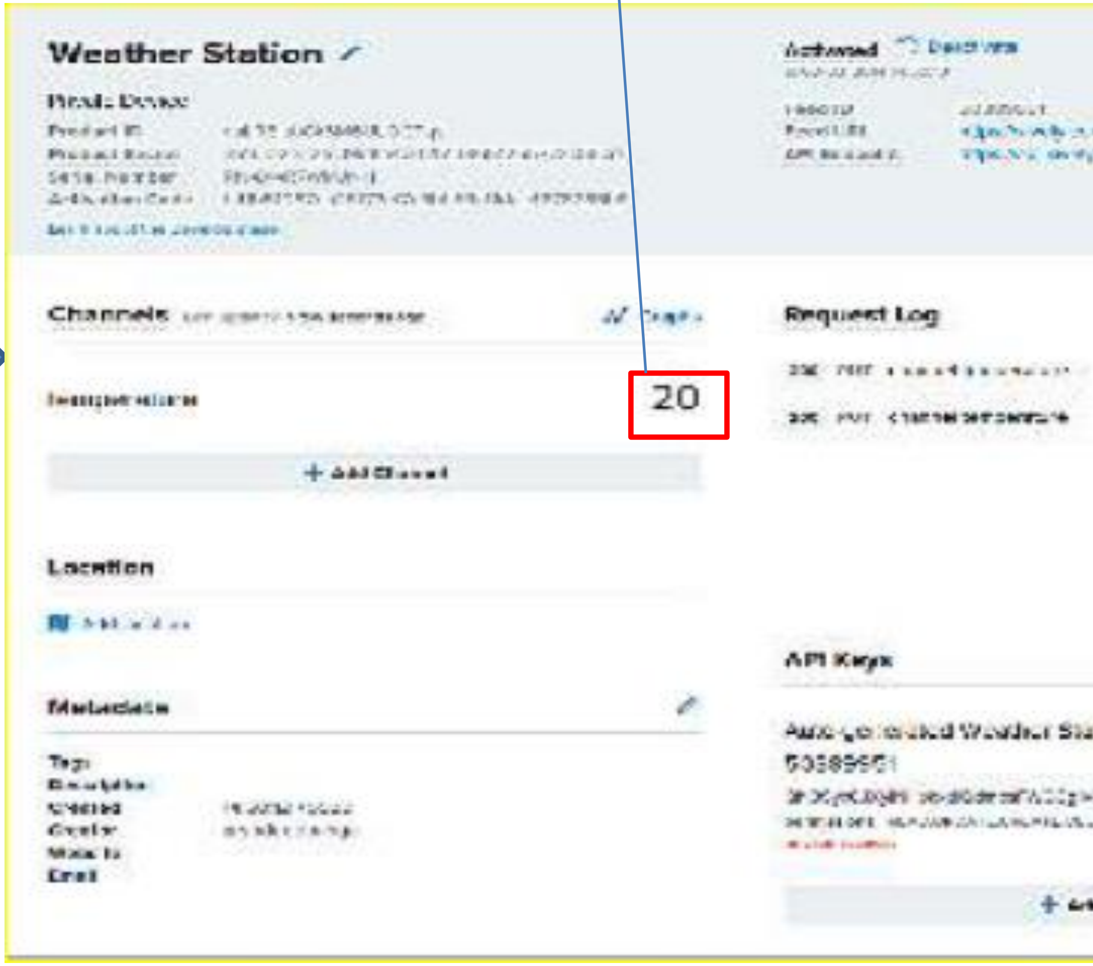
Temperature data is sent to temperature cl
the ***runcontroller()*** function every 10 secor
Xively dashboard.

Siddaganga Institute of Technology

Xively Cloud : Python Program for sending data to a Xively Cloud

Temperature read Data is updated as 20





Weather Station

[illegible]

Channels

Request Log

[illegible] 20

+ Add Channel

Location

API Keys

Metadata

Tags
Description
Created
Creator
[illegible]
Email

Xively Cloud for IoT - references

* http://www.reuk.co.uk/wordpress/raspberry-pi/raspberry-pi-temperature-logger-with-xively/
* https://dzone.com/articles/how-to-use-xively-platform-in-iot-project

http://ww1.microchip.com/downloads/en/Appnotes/Atmel-42275-Connecting-Wireless-Networks-to-the-Internet-using-Xively-Technology_AP-Note_AT07926.pdf

https://embeddedcomputing.weebly.com/data-and-action-iot-with-xively-and-zapier.html

https://github.com/davidmat/iot-temperature-sensor/blob/master/arduinotoxively.py

* https://subscription.packtpub.com/book/hardware_and_creative/9781783986064/5/ch05lvl1sec36/configuring-your-xively-account

https://shodhganga.inflibnet.ac.in/bitstream/10603/107638/9/09_chapter%20-%205.pdf