Experiment-1

Prepare a list of features of above IOT PLATFORM.

1. Connectivity

In the case of IoT, the most important feature one can consider is connectivity. Without seamless communication among the interrelated components of the IoT ecosystems (i.e sensors, compute engines, data hubs, etc.) it is not possible to execute any proper business use case. **IoT devices** can be connected over Radio waves, Bluetooth, Wi-Fi, Li-Fi, etc. We can leverage various protocols of internet connectivity layers in order to maximize efficiency and establish generic connectivity across IoT ecosystems and Industry. There may be special cases where the IoT ecosystem is built on-premises or in an intranet.

2. Sensing

We humans can naturally understand and analyze our circumstances easily based on our past experiences with various things or situations. In the case of IoT in order to get the best of it, we need to read the analog signal, convert it in such a way that we can derive meaningful insights out of it. We use Electrochemical, gyroscope, pressure, light sensors, GPS, Electrochemical, pressure, RFID, etc. to gather data based on a particular problem. For example for automotive use cases, we use Light detection sensors along with pressure, velocity and imagery sensors. To make a use case successful we need to choose the proper sensing paradigm.

3. Active Engagements

IoT device connects various products, cross-platform technologies and services work together by establishing an active engagement between them. In general, we use **cloud computing** in blockchain to establish active engagements among IoT components. In the case of Industry grade, IoT solutions raw analog data need to be acquired, preprocessed and rescale as per business capacity. As per Google, only 50% of structured and 1% of unstructured data is used to make important business decisions. So while designing the IoT ecosystems carriers need to consider the future needs of manipulating such a huge scale of data to satisfy incremental business needs. One can confuse the need of active engagements with scale, practically it means your systems should be able to handle huge data across various technologies, platforms, products, and industries.

4. Scale

IoT devices should be designed in such a way that they can be scaled up or down easily on demand. In general, IoT is being used from smart home automation to automating large factories and work stations, so the use cases vary in scale. A carrier should design their IoT infrastructure depending upon their current and future engagement scale.

5. Dynamic Nature

For any IoT use case, the first and foremost step is to collecting and converting data in such a way that means business decisions can be made out of it. In this whole process, various **components of IoT** need to change their state dynamically. For example, the input of a temperature sensor will vary continuously based on weather conditions, locations, etc. IoT devices should be designed this keeping in mind.

6. Intelligence

In almost every IoT use cases in today's world, the data is used to make important business insights and drive important business decisions. We develop machine learning/ deep learning models on top of this massive data to obtain valuable insights. The analog signals are preprocessed and converted to a format on which **machine-learning models** are trained. We need to keep in mind the proper data infrastructure based on business needs.

7. Energy

From end components to connectivity and analytics layers, the whole ecosystems demand a lot of energy. While designing an IoT ecosystem, we need to consider design methodology such that energy consumption is minimal.

8. Safety

One of the main features of the IoT ecosystem is security. In the whole flow of an IoT ecosystem, sensitive information is passed from endpoints to the analytics layer via connectivity components. While designing an IoT system we need to adhere to proper safety, security measures, and firewalls to keep the data away from misuse and manipulations. Compromising any component of an IoT ecosystem can eventually lead to failure of the whole pipeline.

9. Integration

IoT integrates various cross-domain models to enrich user experience. It also ensures proper trade-off between infrastructure and operational costs.

IoT and Python

- IoT development requires a high-level programming language. A programming language that focuses on code readability.
- Python is the IoT app developers' first choice because of its dynamism and interpretation power.
- Python is easy to learn and it generally has few steps of implementation as compared to Java and C.
- The multi-functional programming language is used for mathematics, software development, system scripting, and web development.
- Used for server-side development. Handles complex data with ease.
- Runs Python on Mac, Linux, Raspberry Pi, and Windows easily.
- A fast programming language that runs on the interpreter.
- It is a Scripting language that enables developers to easily develop web and desktop applications.
- A functional, procedural and object-oriented language that can be translated to a binary language like java.

Role of Python in IoT Development

IoT development requires a database to store generated data. MySQL provides IoT app developers the go-to relational database. It is the most convenient tool that evades the requirements to execute shell commands within a Python script. In the IoT development other programming languages like C, C++, Assembly, Java, JavaScript, and PHP. Python is the developers' favorite programming language when it comes to the development of IoT applications.

- Python has a clear syntax that gives developers the idea of code identification instead of {}
- Its syntax is similar to the English language
- Python has an interpreter system which it runs on. The developers can easily execute the code as soon as it is written. It also facilitates quick prototyping.
- Integration with other languages is possible while using Python. Python app developers can easily put their code in other programming languages including C++, Java, etc.
- An extensible language that enables developers to write programs with fewer lines.
- It has code portability. The developers do not have to write new codes every time for different machines.
- An Open-source framework that is easily available for public download.
- Huge community support enables developers to create user-friendly apps each time.

EXPERIMENT-2

Establish connection between Arduino and bluetooth module.

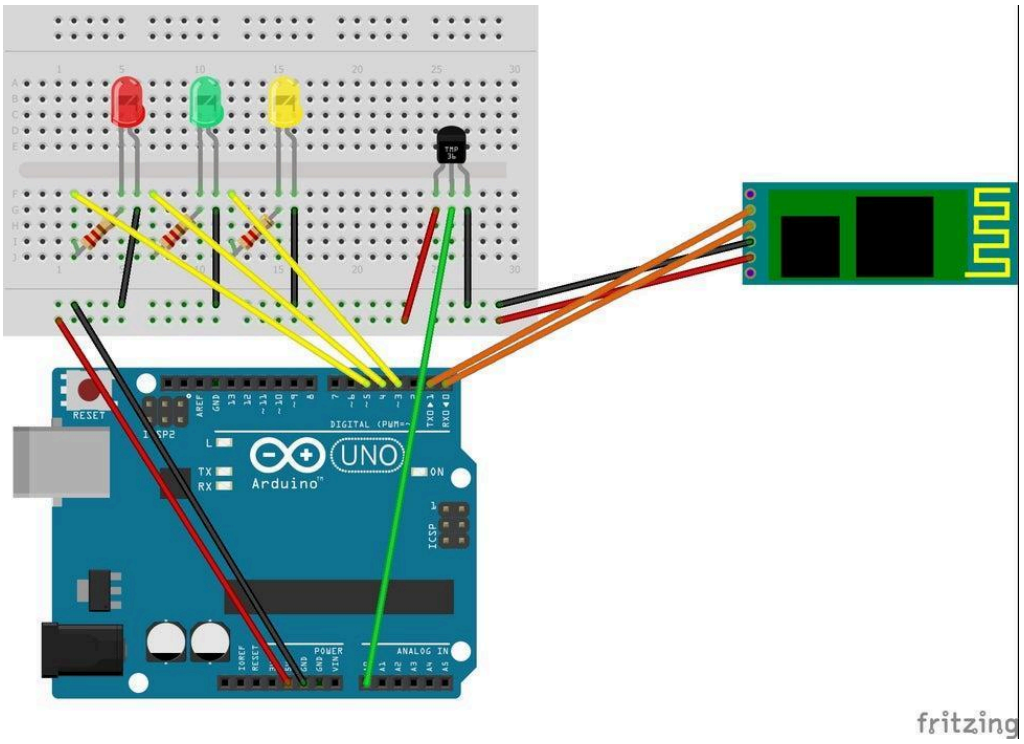
). Arduino listens for commands to light some LED's or show its status. In addition, a timer interrupt makes it check for temperature via a TMP36 sensor: if temperature is greater than a threshold a LED is lit; every n seconds (where n is a parameter set through the app) a status report is sent to the app. A simple command structure enables the app to send parameters and values to Arduino and the other way round.

There are many Bluetooth modules and even Arduino shields: my choice has fallen on JY-MCU that I bought from Hobby Components in the UK.

The JY-MCU is a class-2 Bluetooth module that acts like a serial port with no need of any software configuration on the Arduino. This module is available in several configurations, and this has been the first hurdle to overcome.

The one we need to make the connection between Arduino and the Android phone is a Slave Module. To make things very simple, there are two types of devices: Master and Slave; a Master can communicate with more than one Slave while a Slave can communicate with a single Master at a time, Master-Master and Slave-Slave communication is not allowed. **Since the Bluetooth module in all smartphones is of Master type, the one we need for Arduino must be a Slave** (this has nothing to do with client-server communication as we'll see later in the app description

Step 1: The Circuit



Components:-

1 x Arduino Uno
3 x 220 Ohm Resistrors
1 x Green LED
1 x Yellow LED

1 x Red LED
1 x TMP36 Temperature Sensor
1 x JY-MCU Bluetooth Slave Module (see introduction)
1 x Breadboard
wires

Connections:-

Step 1): Connect Ground and 5V from Arduino to breadboard.
Step 2): Place the LEDs on the breadboard and connect their catodes to ground; connect their anodes to Digital Pins through a 220 Ohm resistor each: Yellow to Pin 3, Green to Pin 4 and Red to Pin 5.
Step 3): Place the TMP36 sensor on the breadboard and connect its pins to 5V, Ground and A0 Arduino Pin.
Step 4): Connect the provided cable to the JY-MCU Bluetooth module on one side and to the breadboard to the other side; connections are as follows:
VCC <--> 5V
GND <--> GND
TXD <--> Pin 0 (Rx)
RXD <--> Pin 1 (Tx)

The sketch will also make use of Arduino built-in LED on Digital Pin 13.

The connections of the Bluetooth module can be a little confusing since TXD goes to Rx and RXD goes to Tx: here's an explanation. Transmit and Receive refer to each device, therefore a transmission coming out of TXD pin of the Bluetooth module must be received by Arduino on the Rx Pin 0; similarly a transmission going out of Arduino Tx Pin 1 must reach the JY-MCU Bluetooth module on its RXD pin.

Warning: The Bluetooth module may interfere with PC to Arduino communication: disconnect VCC when programming the board. (in my tests this has not been the case, but for sure it won't do any harm).

connect your Arduino to the internet by adding ESP8266 WiFi Module!

The ESP8266 Wifi module is a complete WiFi network where you can easily connect as a serving Wi- Fi adapter, wireless internet access interface to any microcontroller based design on its simple connectivity through Serial Communication or UART interface.

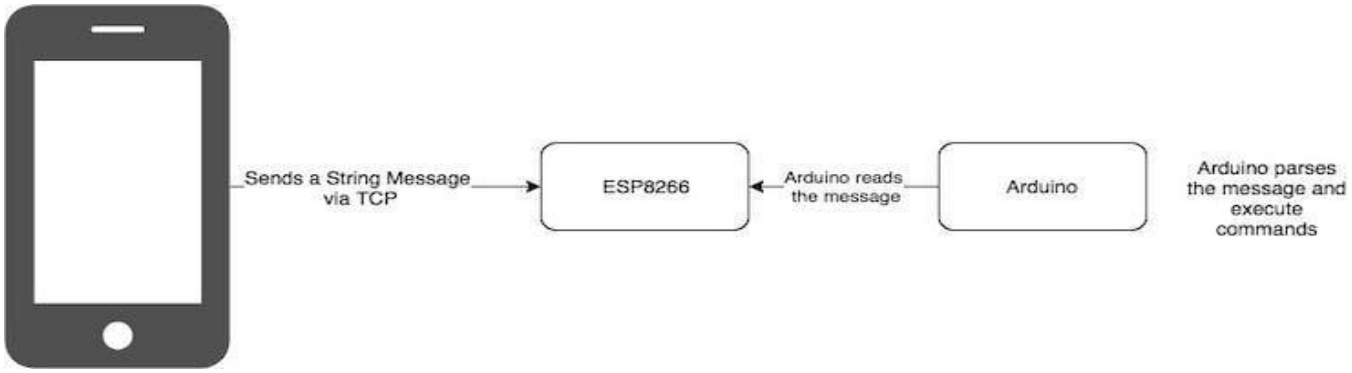
Adding this module to your Arduino UNO will open you to more and exciting projects.

What is the process?

There are a lot of ways to use ESP866 for communication. Some may use it to send / receive data online or regularly upload data. **For this tutorial** I will just show you how can we communicate to Arduino wirelessly using your phone (Android or iPhone) . This will be done offline so no need to have internet connection.

Our ESP8266 will serve as Access Point (AP Mode), meaning it will provide access to Wi-Fi network to other devices (stations) and connects them further to a wired network

The process it pretty simple. Use your phone to send any command to Arduino. With the help of ESP8266, everything will work wirelessly.



Communication Process

Upload Sketch

First of all, upload my sample Arduino sketch.

If you want to know whats on my sketch go to **About the code section**

You can download the attached sketch or get the updated version on my **github repo.**

If you don't know how to upload a sketch then **Read here**

Do I need to flash my ESP?

No. Just use the stock firmware. But in case you need then read here

Built Circuit

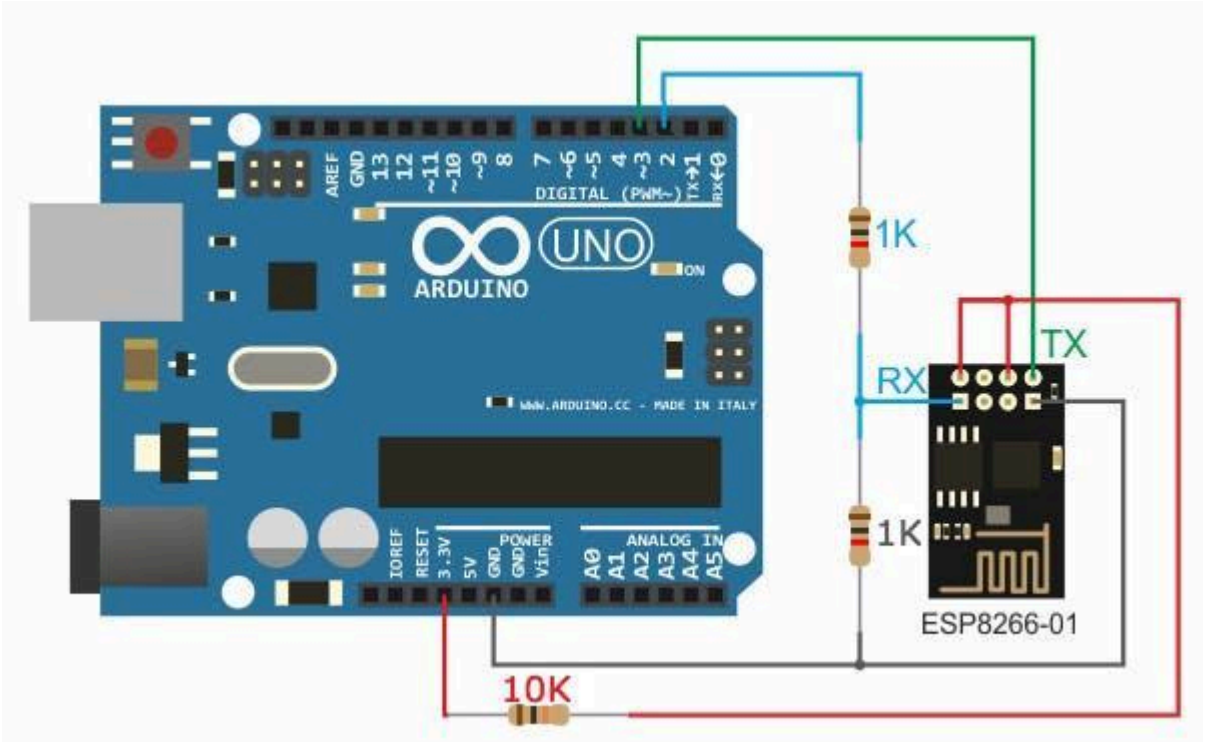
Connect the pins more described on this Pin table

ESP8266	Resistor	Arduino Uno
GND		GND
VCC/3.3V/Power	10K	3.3V
CH_EN/Enable	10K	3.3V
TX		PIN 3
RX	1K	PIN 2
RX	1K	GND

Follow these steps.

- connect both ESP's VCC/3.3V/Power Pin and Enable Pin (red wires) to 10K resistor then to Uno's +3.3V power pin.
- connect ESP's Ground/GND Pin (Black Wire) to Uno's Ground/GND Pin.
- connect ESP's TX (green wire) to Uno's Pin 3

- cconnect ESP's RX (blue wire) to 1K resistor then to Uno's Pin 2.
- cconnect ESP's RX (blue wire) to 1K resistor then to Uno's GND Pin.



Set up connection

Once everything is set up, you would notice that your ESP8266 Wifi will be available within the range of your phone.

1. Download TCP Client for Android

You can download any TCP Client available in Play Store but I used TCP Client by Sollae Systems

2. From your phone, Connect to your ESP8266 Wifi

If your ESP8266 wifi is not displayed from available wifi networks, make sure your Arduino is running and everything is connected correctly. If not troubleshoot your ESP by following its documentation.

Usually the name of the wifi / ssid will start in **ESP** following its version name, mine is **ESP11.**

3. Once connected, get the Static IP address.

The default IP Address in AP mode is 192.168.4.1 .

You can change the static IP by following this Wifi.config() reference.

EXPERIMENT-3

IoT based Humidity and Temperature Monitoring System using Arduino Uno

internet. The devices may vary from the temperature measuring equipment and vehicles SOS system to other electronic devices such as sensors, softwares, and network connectivity facilities, which sanction collecting and exchanging data. The twentyfirst century has witnessed a massive paradigm shift to and temperature sensors can use [6]. Resistors, semiconductors, thermistors estimate. Temperature and

humidity are vital in observing andunderstanding nature. IoT comes into the picture here by significantly enhancing the efficiency of the mechanism and systematically cutting down on human involvement, and Practically, every part of exercise contains controlled schedules of temperature as well as humidity. , the exact value of temperature with its significant feature in any field is essential in monitoring [5]. Constant perception in temperature is utilized in various industries like the pharmaceutical industry as the driving force behind these monitoring systems, computerized and straightforward temperatures values. These components are present inside the sensor to retrieve the temperature in consonance with the circumstances. The primary goal of our system is to supervise the live temperature and humidity within alow cost [7]. Raspberry Pi is the observational system or controller which is used for the cloudsaving. Python is the programming language whichis utilized in Raspberry Pi. HTU 211D sensors is a temperature sensor which is used here for the sensing

HARDWARE REQUIREMENTS

- Power supply
- Lcd display
- Arduino
- Temp senso
- Humidity sensor
- Ultrasonic senso
- Wifi module

SOFTWARE REQUIREMENTS

- Arduino IDE
- Embedded C
- Proteus

WORKING:

The Temperature and Humidity Sensor Project will be controlled using an application named FIRE BASE CONTROL (Available for Android and windows) using Arduino, an Ethernet Shield and its libraries. User can securely login over Fire base control to control and monitor the room temperature and humidity. The code includes: The Arduino sketch. The Arduino sketch handles the interchangesby setting up the system. The sketch runs the program and conveys each line in turn over the server. Clients can login remotely on this web server. Utilization of DHT22 Sensor: DHT22 capacitive moistness detecting computerized temperature and dampness module is one thatcontains the intensify that has been adjusted carefully to flag yield of the temperature and stickiness sensors. The sensor incorporates a capacitive sensor, wet parts and a high precision temperature, estimation gadgets, and associated with a superior 8-piece microcontroller. The item has great quality, quick reaction, solid enemy of

sticking capacity, and signific expense. It is certainly dependable and has incredible perseverance. fromFirebase and glue it into the line the program the Hostname and Password with your Fig (1) Block

Diagram of Monitoring system The Temperature and Humidity Sensor Project will be controlled using an application named FIRE BASE CONTROL (Available for Android and windows) using Arduino, an Ethernet Shield and its libraries. User can securely login over Fire base control to control nd monitor the room temperature and humidity. The code includes: The Arduino sketch. The Arduino sketch handles the interchanges by setting up thesystem. The sketch runs the program and conveys each line in turn over the server. Clients can login onthis web server. Utilization of DHT22 Sensor: DHT22 capacitive moistness detecting computerizedtemperature and dampness module is one that contains the intensify that has been adjusted carefully to flag yield of the temperature and stickiness The sensor incorporates a capacitive sensor, wet parts and a high precision temperature, estimation gadgets, and associated piece microcontroller. The item has great quality, quick reaction, solid enemy of sticking capacity, and significant expense. It is certainly dependable and has from Firebase and glue it into the line the program. Next, supplant WIFI name and Wi-Fisecret key in the two lines given beneath in the program.At that point assemble the program and transfer to an Arduino Uno through Arduino IDE. Guarantee that WiFi modem and web associationin your Smartphone or PC are working appropriately. After transferred a program, theTemperature and Humidity information is transferred on Firebase Console stage. You can see it graphically in the private view window ofyou divert as appeared in Fig: Also, you can readyto see the transferred informationfrom sequentialport of Arduino IDE. Circuit outline for observing moistness and temperature is appeared in Fig. 2. It is worked around Arduino MCU, DHT11 sensor and ESP8266 Wi-Fi module. In this task, we will construct a little circuit to interface Arduino MCU with DHT11 Temperature and Humidity Sensor. One of the fundamental utilizations of interfacing DTH11 sensor with Arduino is climate observing. All the DHT11 Sensors are precisely adjusted inthe research facility and the outcomes are putaway in the memory.A solitary wire correspondence can be set up between any microcontroller like Arduino and the DHT11Sensor. Also, the length of the link can be up to

RESULT:

The code is uploaded in the Arduino. software application and the Arduino MCU board were configured. In the Firebase Console the account will be created then the Data base will be created. According to the data base the Threshold values are inserted to the required temperature values and the humidity values .This make the values as a barrier when the humidity crosses its limit in a room atmosphere the notifications will be alerted and the with the google account when we logon to the account the values are displayed.

Experiment No.04

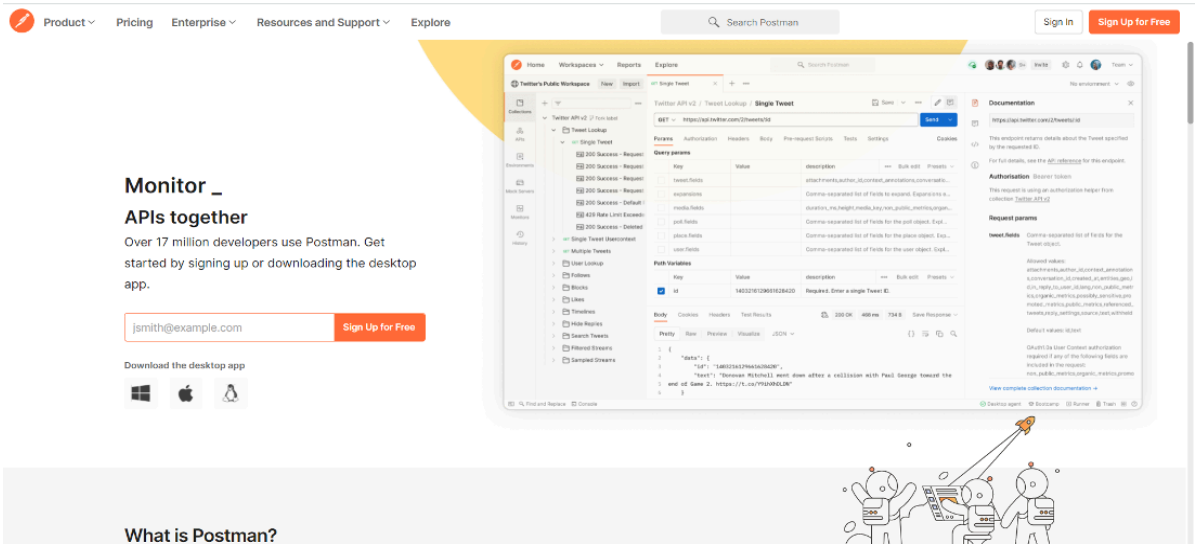
How to Download and Install Postman on Windows?

Postman is a platform for building and using APIs and helps for simplifying the steps in the APIs lifecycles to streamline collaboration for creating faster APIs. It includes various API tools to accelerate the development cycle, including the design mockups and testing documentation, etc. Postman will use it directly on a web browser or we can also download the desktop version also for convenient use. Nowadays postman is also used by multinational companies like Twitter, Gear4music, BetterCloud, Momentive, etc. It also offers an API through which users can access data on the platform. It offers features such as search, notifications, alerts, security warnings, etc.

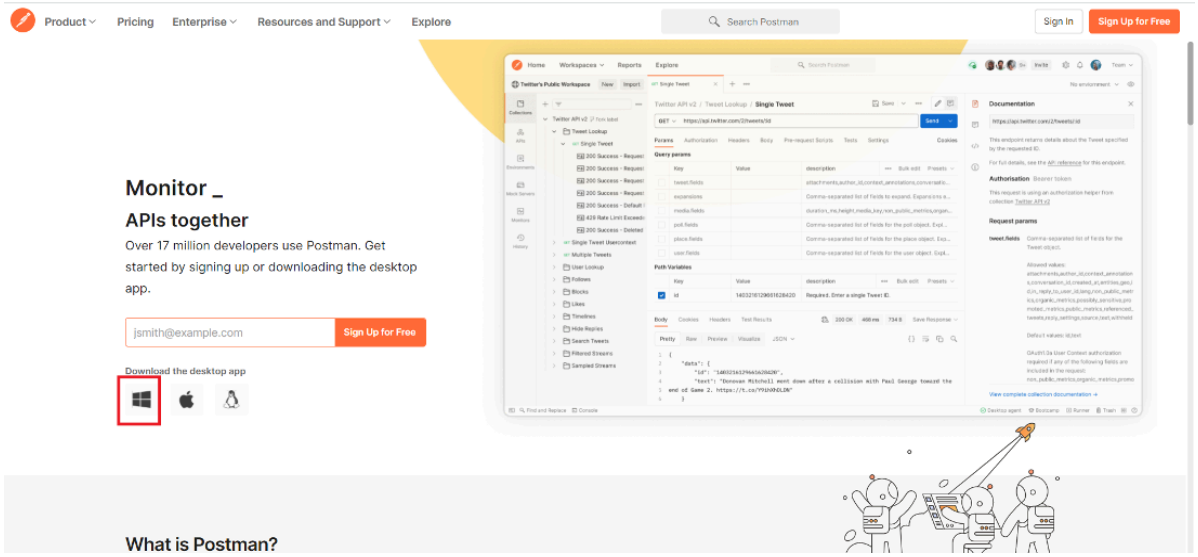
Installing Postman on Windows

Follow the below steps to install Postman on Windows:

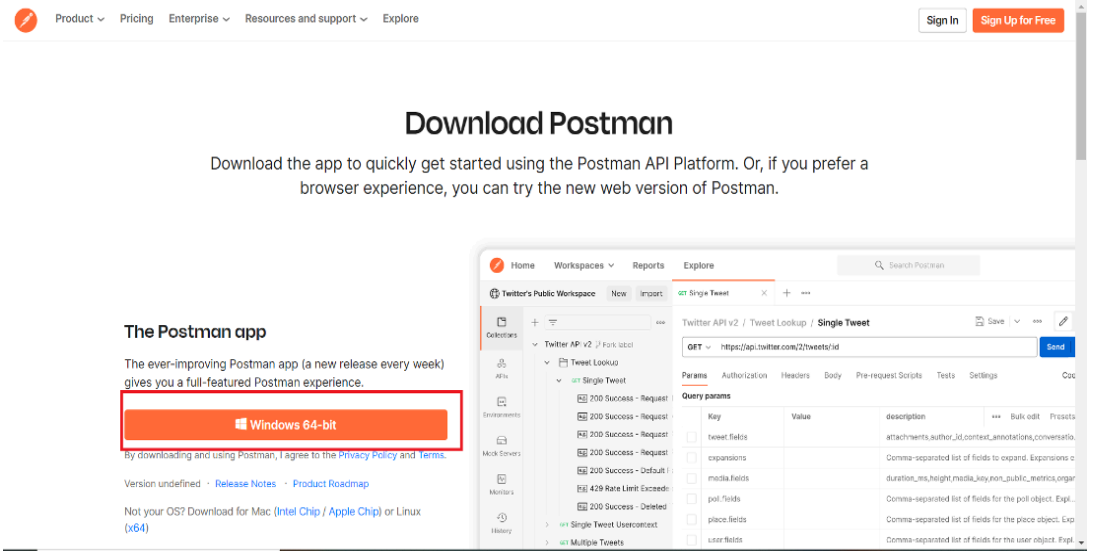
Step 1: Visit the https://www.postman.com/ website using any web browser.



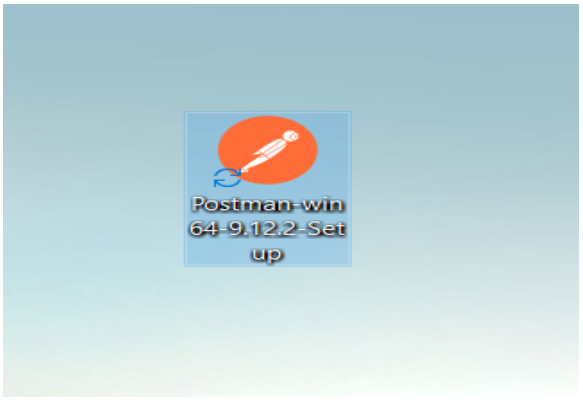
Step 2: Click on Windows Button to download.



Step 3: Now click on Windows 64 – bit button.

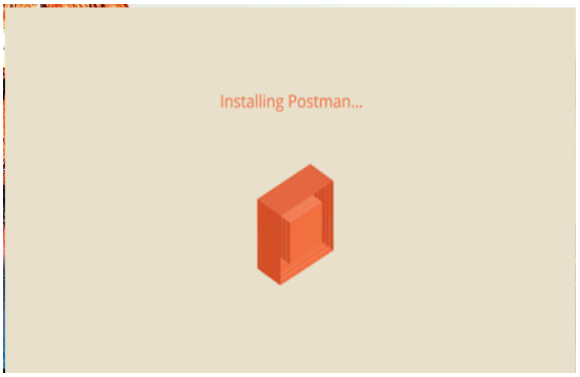


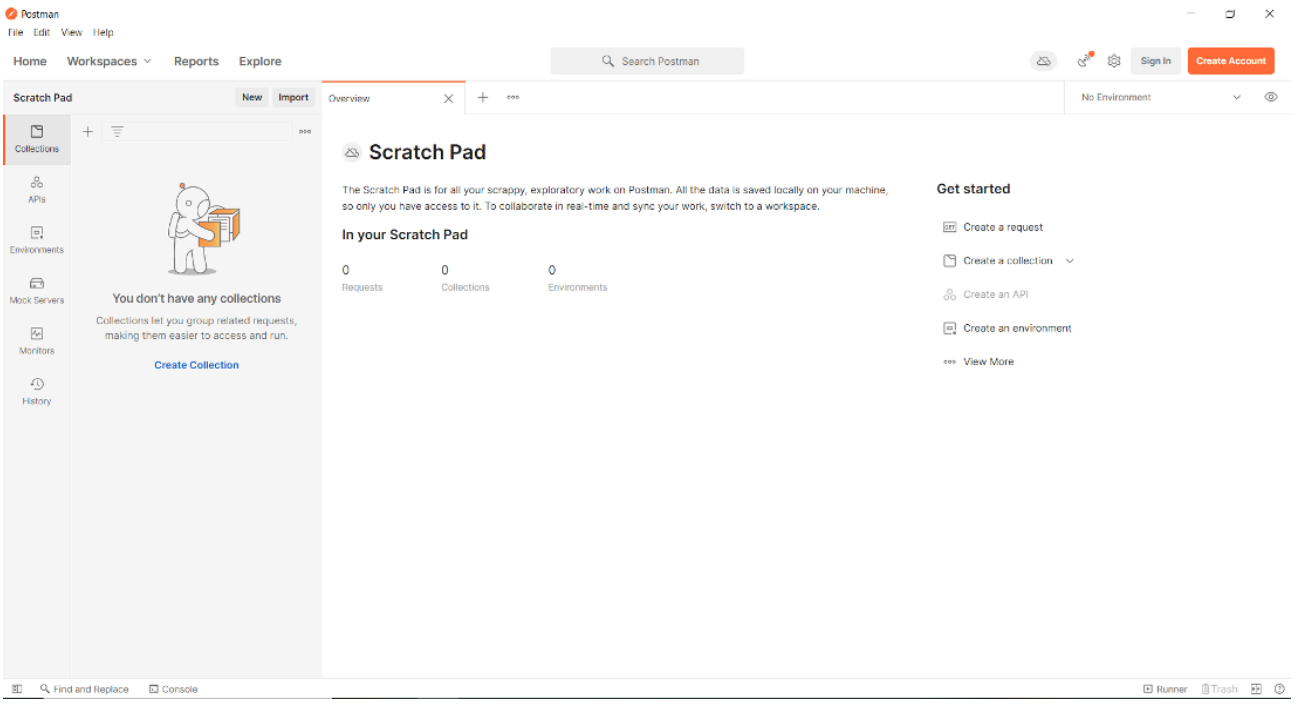
Step 4: Now check for the executable file in downloads in your system and run it.



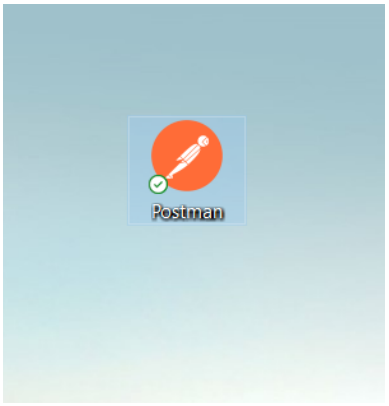
Step 5: Now installing process will start it takes a minute to install in the system.

Step 6: After installing the program the software opens automatically. Now you can see the interface of the software.





- Postman is successfully installed on the system and an icon is created on the desktop.



- Congratulations!! At this point, you have successfully installed **Postman** on your windows system .

4.2 REST API & test using POSTMAN

REST is an acronym for **Representational State Transfer**. It is an architectural style for distributed hypermedia systems.

An API is an **application programming interface**. It is a set of rules that allow programs to talk to each other. The developer creates the API on the server and allows the client to talk to it.

REST determines how the API will look. It is a set of rules that developers follow when they create their API. One of these rules states that you should be able to get a piece of data (i.e., a resource) when you hit a specific URL.

The key abstraction of information in REST is a **resource**. Any information that can be named can be a resource:

- a document
- an image
- a temporal service
- a collection of other resources
- a non-virtual object (e.g., a person)

REST uses a resource identifier to identify the particular resource involved in an interaction between components.

Another important thing associated with REST is resource methods, which perform the desired transition. REST APIs enable us to develop any kind of web application that has all possible **CRUD (create, retrieve, update, delete)** operations.

HTTP methods

The suitable HTTP method for the action performed by API are:

- GET
- PUT
- POST
- DELETE

GET:
GET requests only retrieve resource representation/information – they do not modify it in any way. Since GET requests do not change the state of the resource, these are said to be safe methods.

PUT:
PUT APIs are primarily used to update existing resources (if the resource does not exist, then the API may or may not decide to create a new resource). If a new resource has been created by the PUT API, the origin server MUST inform the user agent via the HTTP response code

POST:
POST APIs create new subordinate resources (e.g., a file is subordinate to a directory containing it and a row is subordinate to a database table). Strictly talking in terms of REST, POST methods are used to create a new resource in the collection of resources.

DELETE:
DELETE APIs are used to delete resources (identified by the Request-URI).

POSTMAN

Postman is a scalable API testing tool that quickly integrates into the CI/CD pipeline. It started in 2012 as Abhinav Asthana's side project to simplify API workflow in testing and development.

Before working with the testing of API using POSTMAN, install the application from this link.

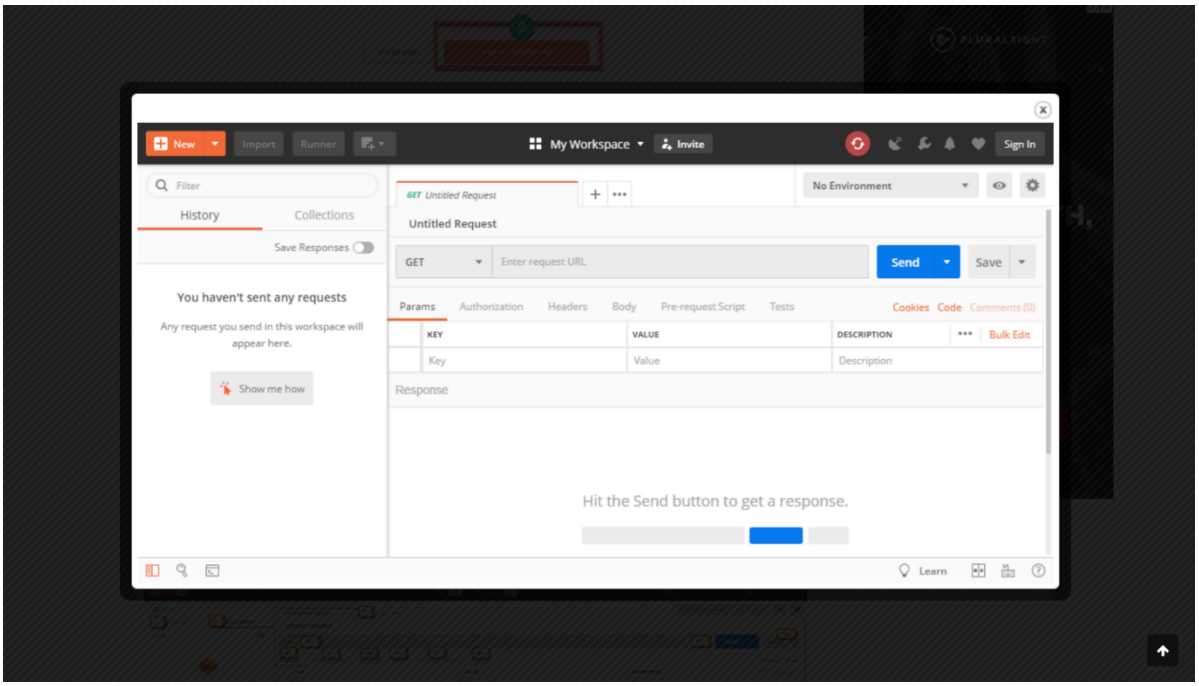
After installation, the POSTMAN application interface can be used for reading, writing, and deleting the contents in API using client HTTP.

POSTMAN

Postman is a scalable API testing tool that quickly integrates into the CI/CD pipeline. It started in 2012 as Abhinav Asthana's side project to simplify API workflow in testing and development.

Before working with the testing of API using POSTMAN, install the application from this link.

After installation, the POSTMAN application interface can be used for reading, writing, and deleting the contents in API using client HTTP.

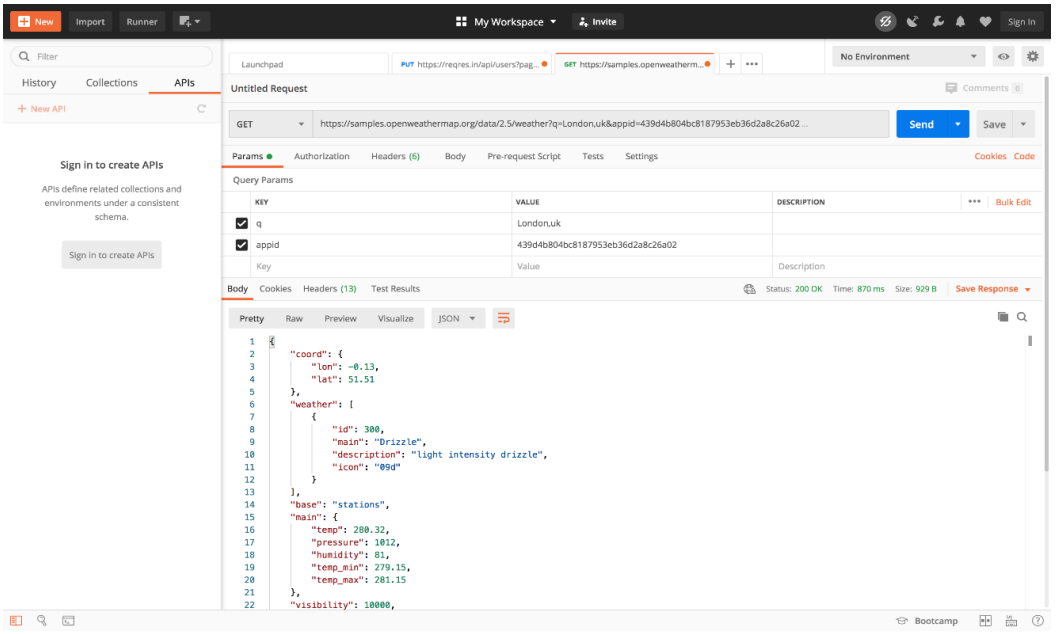


POSTMAN Interface

Let's do an example to illustrate the working of a GET request in POSTMAN. Consider a weather API data and access the data from the API source.

The API source link is : OpenWeatherMap.

The data returned from the OpenWeatherMap is in JSON format.



GET Request in POSTMAN for a arbitrary weather API

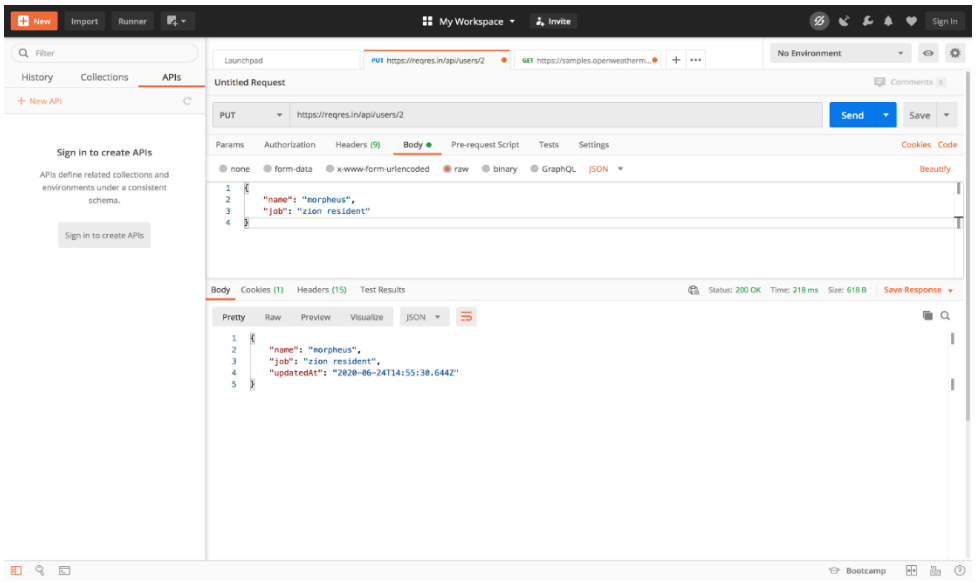
Let us consider another example where we try to PUT data in some API.
PUT requests are different from Get requests as there is data manipulation with the user adding data to the endpoint.

Let us consider an API link where we can PUT the data into the API source.

The JSON data to be PUT into the source API is:

```
{
   "name": "Indrajeet",
   "job": "zion resident"
}
```

After the execution of the command, the JSON data is successfully inserted into the source API.



PUT request in POSTMAN

We can similarly delete the data from the source API using the DELETE request in POSTMAN

What is SOAP?

SOAP, which stands for Simple Object Access Protocol, is an XML-based messaging standard for handling structured data that was created by Microsoft in 1998.

A typical SOAP message consists of four blocks:

1. Envelope: The entirety of a SOAP message is wrapped in the <Envelope> block.
2. Header: This is an optional section inside the <Envelope> which contains information that should be processed before the <Body> section. Authentication credentials are one example of what may be defined within the <Header> section.
3. Body: The body contains the actual payload of the SOAP message. For example, if we're using SOAP to issue remote procedure calls (RPCs), we would likely use the <Body> section to define the RPC call that we're attempting to make, along with any parameters that should be sent along.
4. Fault: The <Fault> block contains information on any errors that occurred while processing a SOAP message.

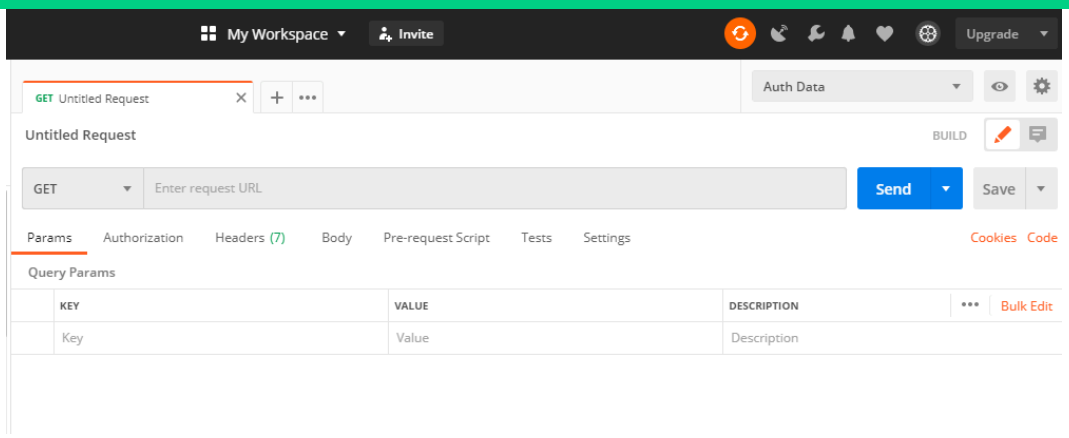
Testing SOAP APIs in Postman

If you're building or consuming SOAP services, you should be thinking about how to efficiently write tests against it. Postman is a popular application for building, testing, and maintaining APIs that was created in 2012 by Abhinav Asthana, and is what we'll be using to test out a sample SOAP service. We'll briefly walk through how to install Postman, and then discuss how to use it to interact with a SOAP service.

Steps to test SOAP APIs in Postman

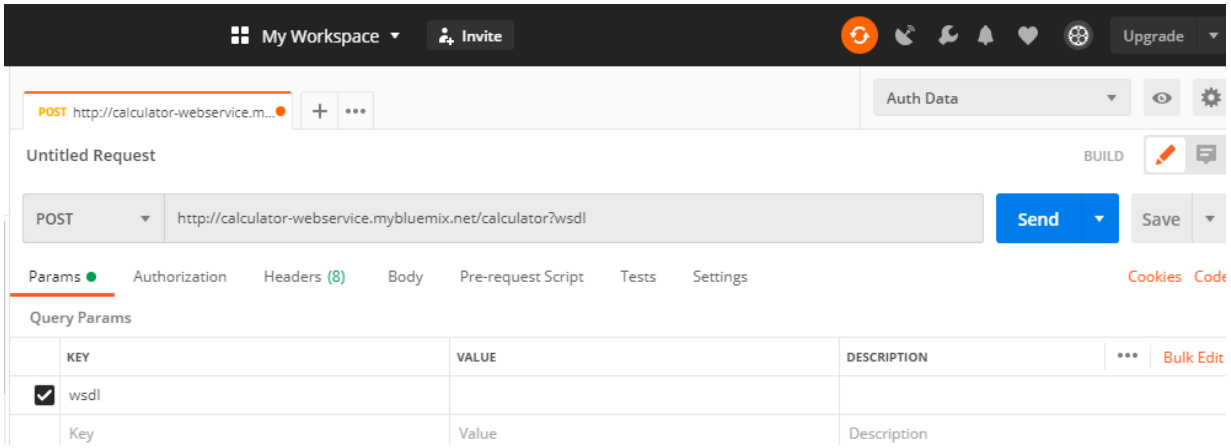
Launch Postman tool

Make sure the Postman tool is installed in our system. If not then download it from here. Launch it after downloading and installing,



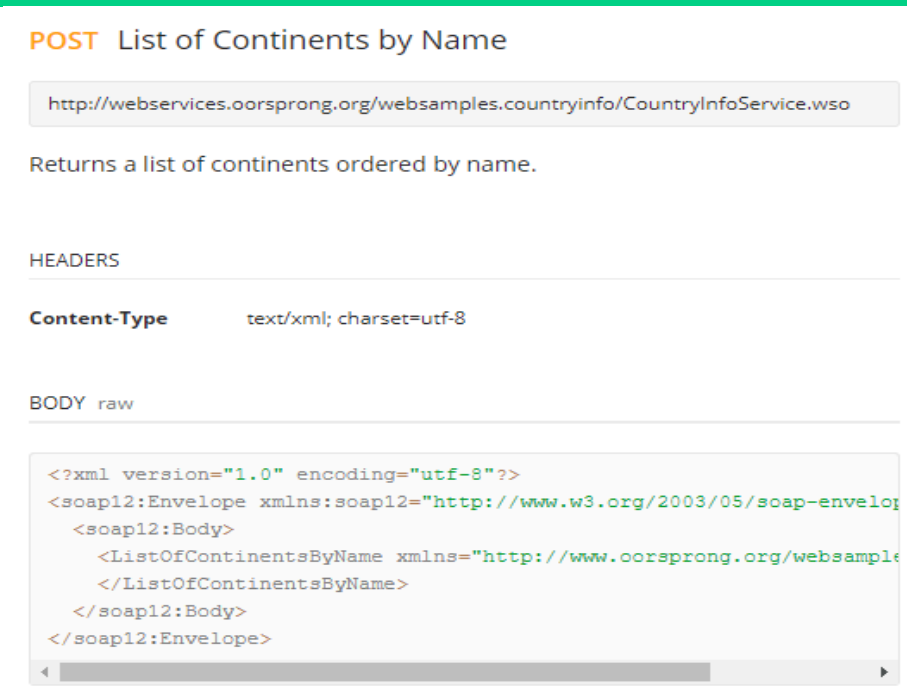
Enter SOAP API URL in the address field of request builder section

Let's use Calculator WSDL for this demo. Copy and paste Calculator WSDL in the address bar as below. Select the HTTP method as POST as mostly SOAP APIs use POST. It may be GET also which can be confirmed from the API documentation which is provided by APIs developers.



Enter request body in XML

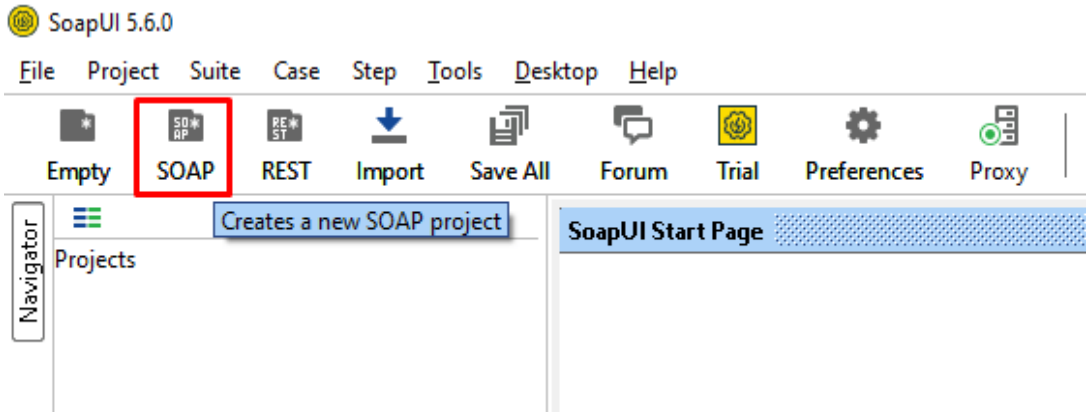
Since APIs based on SOAP supports XML as a message format, we need to pass XML request body. Ideally, you should get an API document where the expected request body of an API will be given. You can see that information is given in demo SOAP APIs provided by Postman here. It gives information like supported HTTP method, required headers, and expected request body.



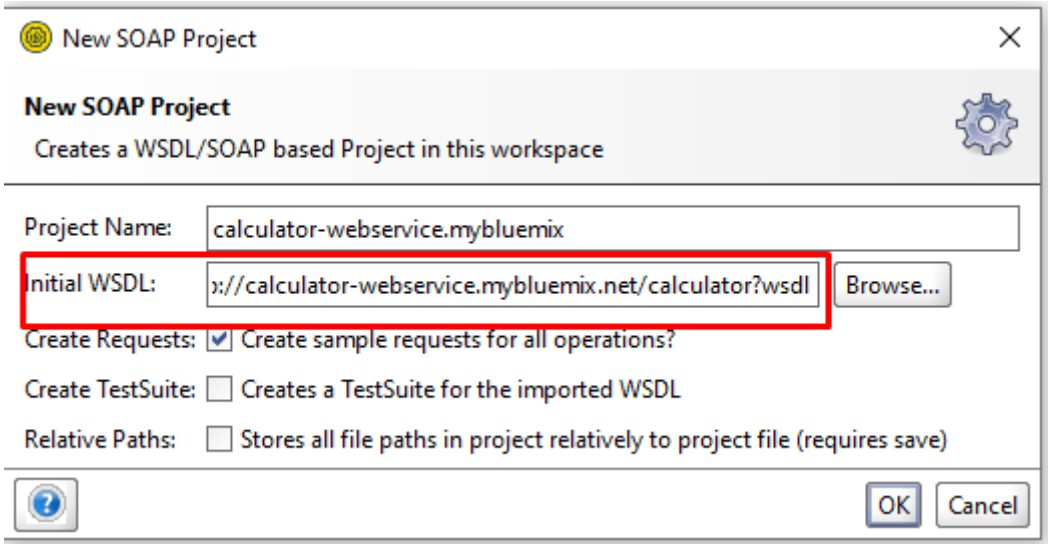
But for Calculator APIs we do not have any documents. Sometimes a developer will share only the WSDL file. In that case, you can use SOAP UI tool to know operations supported by the WSDL file and their request body and other information about APIs.

You can download and install SOAP UI Open Source from the link. installation is very easy like any other normal software. Once installed launch it.

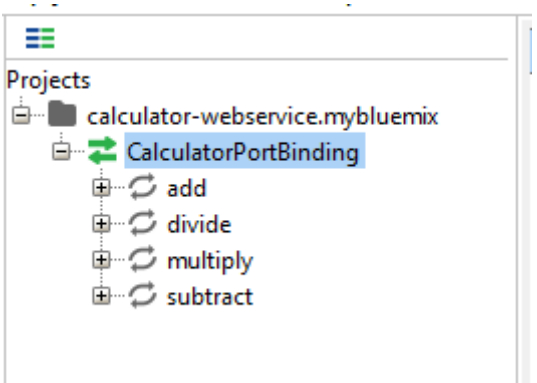
Click on "SOAP" to create a new SOAP project.



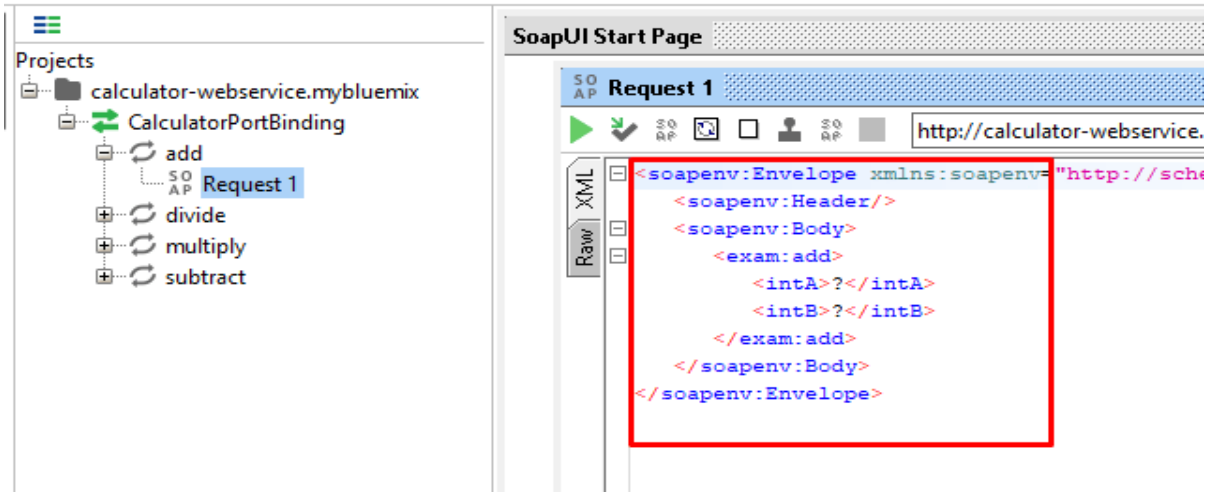
*Enter WSDL in "**Initial WSDL" field.** It will fetch the project name automatically. Make sure you have checked "**Create sample requests for all operations**". It is by default checked. Click on "**OK**".*



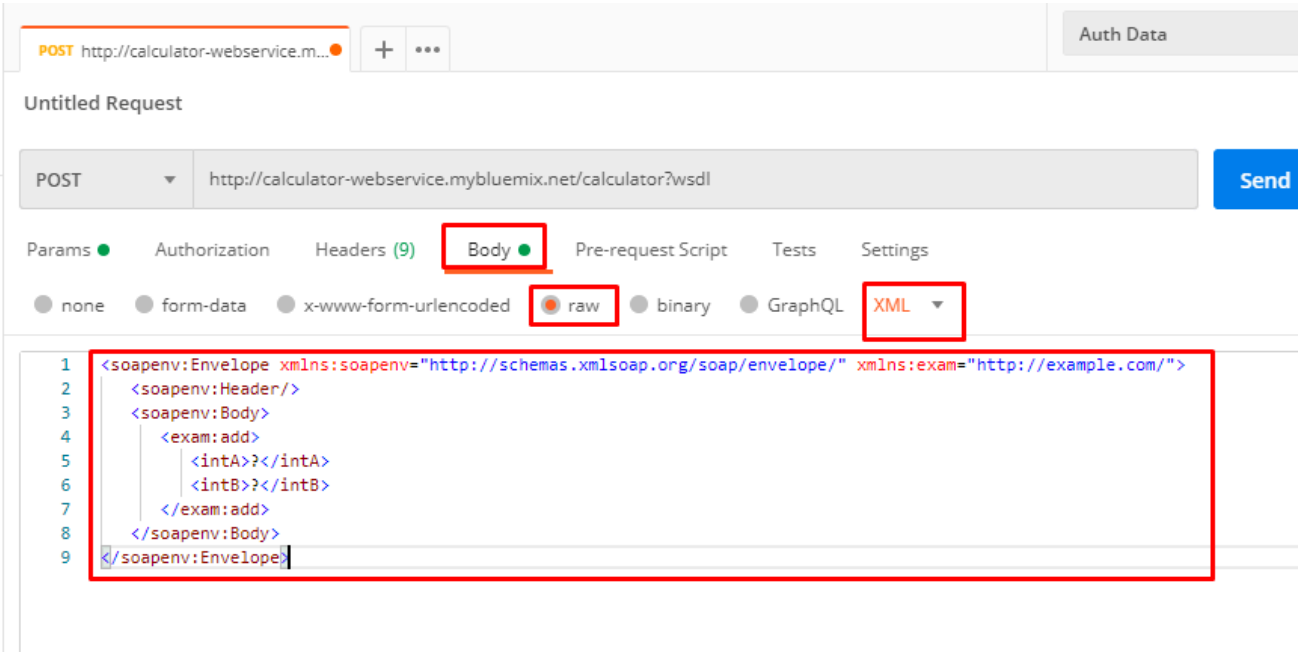
You should see the project will be loaded with all possible operations from WSDL i.e. add, divide, multiply, and subtract.



*When we expand any operation we will see the request body. To see request body click on "**Request 1**".*



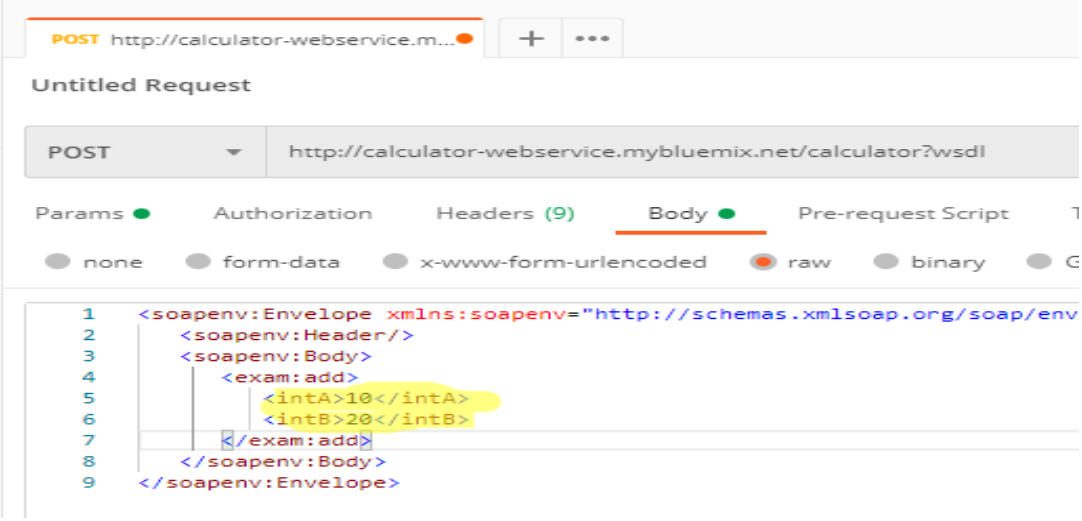
Copy whole XML request body and paste in Postman request body.



*You may be thinking we copied request for "**add**" operation from SOAP UI then there should be a different URI for that operations as well. It is generally followed in REST APIs but in SOAP, WSDL has all the information on supported operations. So which operations will be performed that depends upon the request body. If you observe above XML request body carefully, you can see **<exam:add>**. It represents an "**add**" operation. Similarly, you can see different request bodies in SOAP UI and observe the difference.*

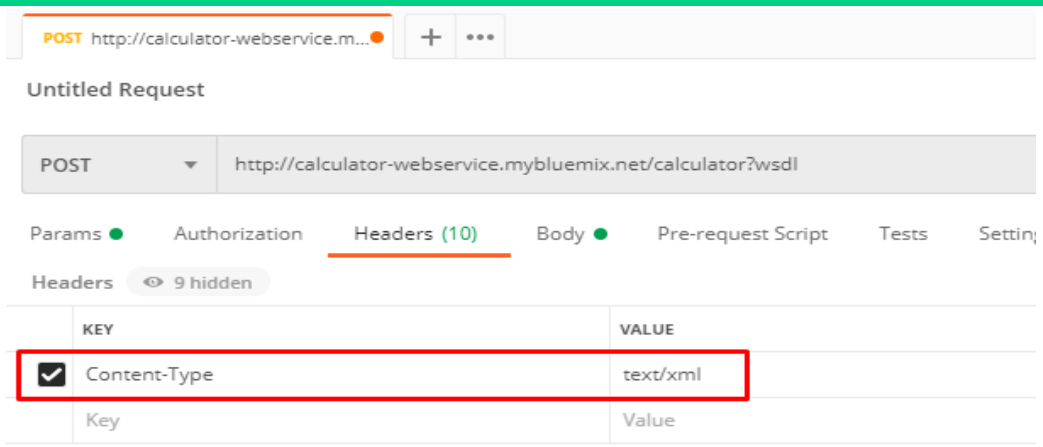
Set input values in XML request body

Once we get the expected request body then we need to fill in required values. For the above SOAP API, we need to pass values of intA and intB which we see in XML body. You need to be aware of functionalities and expected input values of APIs.



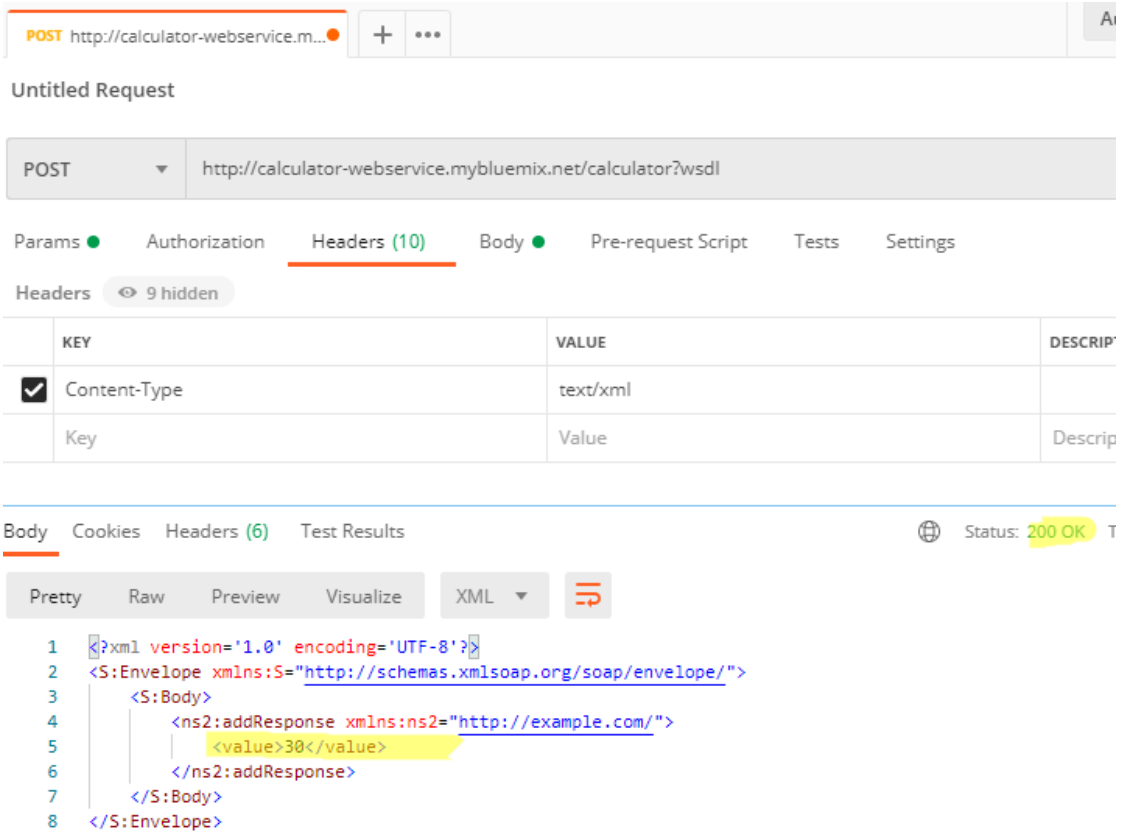
Set header values for the request

*Generally, we need to set Content-Type for SOAP APIs as **text/xml**. This information you can get from API documentation as well. Remember for older SOAP APIs or REST APIs, application/xml will not work.*



Hit the "Send" button

*When we hit the "**Send**" button, we will get an XML response for this API. A response may be a normal text or different format other than XML as well. That all depends on APIs. You can see the status code as well.*


POST http://calculator-webservice.m...
Untitled Request
POST
http://calculator-webservice.mybluemix.net/calculator?wsdl
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
Headers
9 hidden
KEY
VALUE
Content-Type
text/xml
Key
Value
Body
Cookies
Headers (6)
Test Results
Status: 200 OK
Pretty
Raw
Preview
Visualize
XML
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:addResponse xmlns:ns2="http://example.com/">
<value>30</value>
</ns2:addResponse>
</S:Body>
</S:Envelope>