



CLASS OF 2020

---

SNMP  
Platform

---

JUNE • 30 • 2020

## Table OF Content

ABSTRACT .....	2
CH01: DATABASE .....	4
1.0 Legend.....	4
1.1 SNMP ERD .....	5
1.2 Schema.....	6
1.2.1 Tables .....	6
1.2.1. 1 Table: public.users .....	6
1.2.1. 2 Table: public.nodes .....	7
1.2.1. 3 Table: public.action.....	8
1.2.1. 4 Table: public.alarms .....	9
1.2.2 Functions.....	10
1.2.1. 1 Function: public. updatestatus .....	10
CH02: Web Interface.....	12
2.0 Web Interface Screen Shot .....	12
2.1 Web Interface Features .....	13
CH03: SNMP Client & Server .....	15
3.0 What is snmp?.....	15
3.1 What does SNMP do? .....	15
3.2 snmpTerminology .....	15
3.3 What is MIB (management information base)? .....	16
3.4 OID .....	17
3.5 Snmp Operations .....	17
• 3.6 Trap .....	18
CH04: Make Action .....	20
• 4.0 Sending SMS using Nexmo.....	20
• 4.1 What is Nexmo? .....	21
• 4.2 Who uses Nexmo? .....	21
• 4.3 Sending an Email using the JavaMail API. ....	22
• 4.4 Run a Shell Script in Java.....	25

## ABSTRACT

---

### **Simple Network Management Protocol (SNMP)**

Is a protocol for monitoring and controlling communication devices in a network using the Management Information Base (MIB). The device supports SNMPv1 and SNMPv3 security-enhanced version. You can enable SNMPv1 or SNMPv3, or both at the same time. Set settings for each version as appropriate for your network environment and purpose of use.

### **We used two operation:**

#### **1- Get Request**

SNMP manager sends this message to request data from SNMP agent. It is simply used to retrieve data from SNMP agent. In response to this, SNMP agent responds with requested value through response message.

#### **2- Trap**

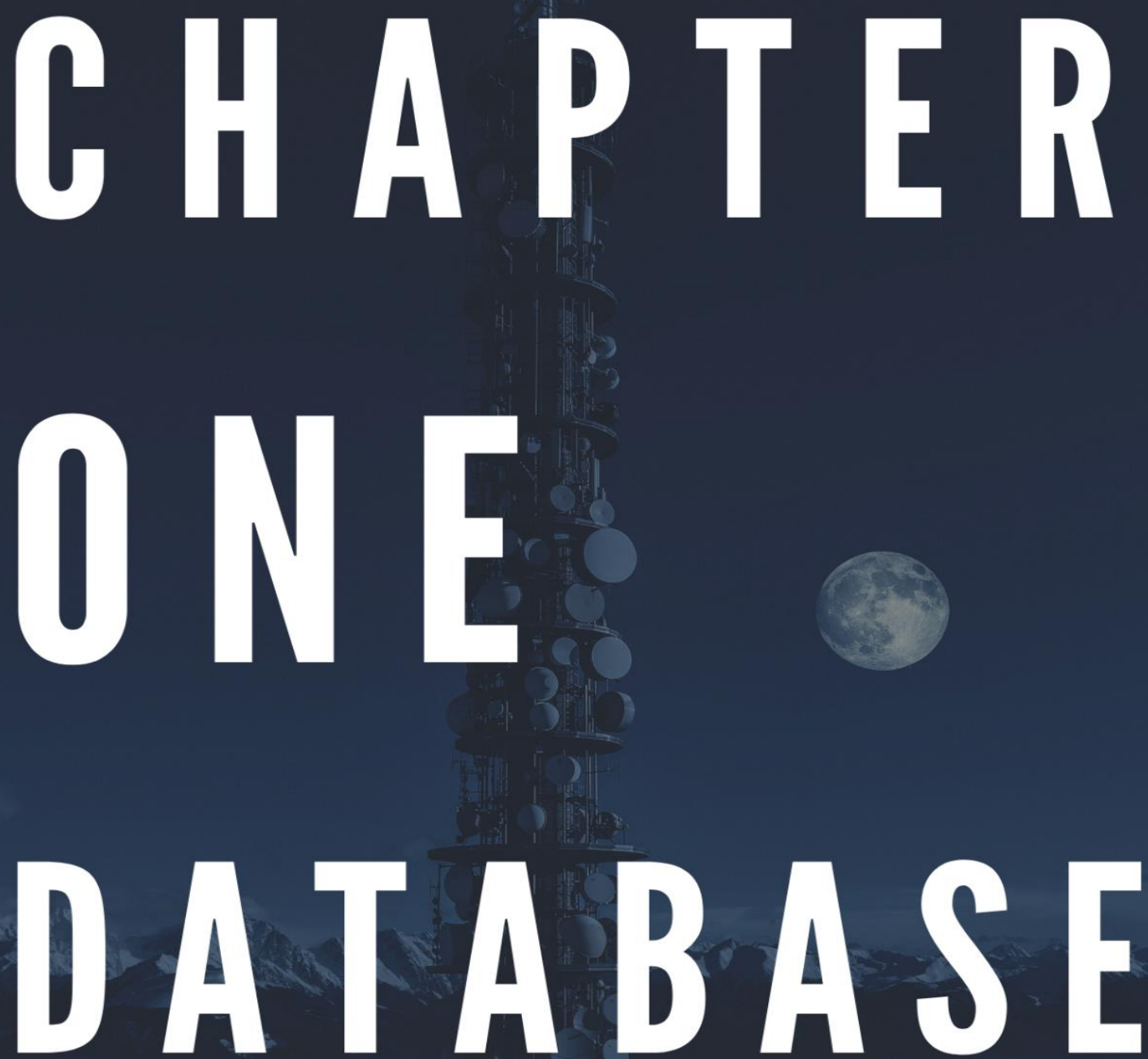
these are the message send by the agent without being requested by the manager. It is sent when a fault has occurred.

When trap Triger

1. Either automatically without the intervention of the admin.
- 2- With the intervention of the admin through Java web application.

Action types to solve a trap:

- 1- Sending SMS using Nexmo
- 2- Sending an Email using the JavaMail API.
- 3- Run a Shell Script in Java.
























# CHAPTER ONE DATABASE

SNMP PLATFORM

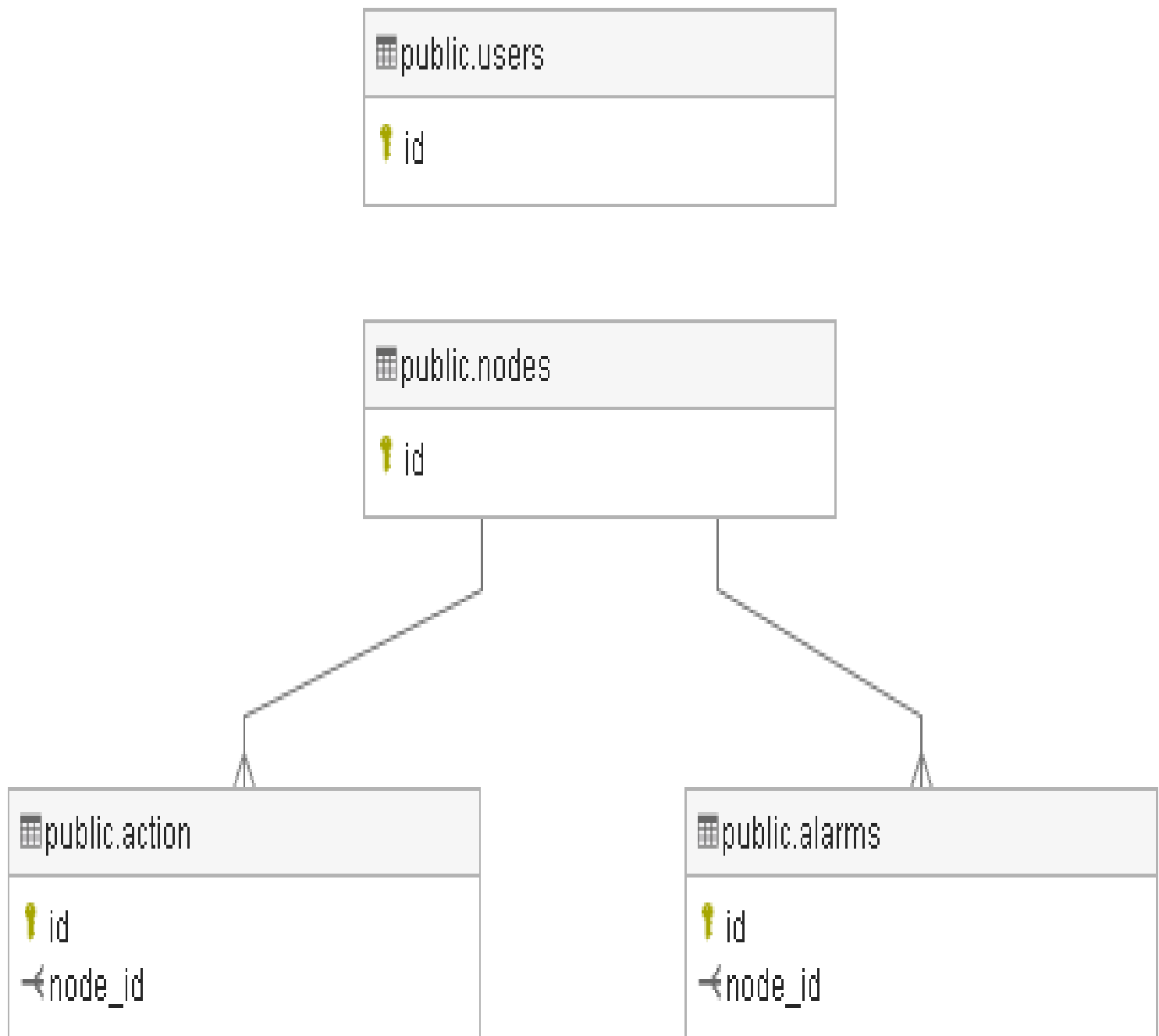
## CH01: DATABASE

---

### 1.0 Legend

-  Primary key
-  Primary key disabled
-  User-defined primary key
-  Unique key
-  Unique key disabled
-  User-defined unique key
-  Active trigger
-  Disabled trigger
-  Many to one relation
-  User-defined many to one relation
-  One to many relation
-  User-defined one to many relation
-  One to one relation
-  User-defined one to one relation
-  Input
-  Output
-  Input/Output
-  Uses dependency
-  User-defined uses dependency
-  Used by dependency
-  User-defined used by dependency

## 1.1 SNMP ERD










## 1.2 Schema


### 1.2.1 Tables

#### 1.2.1. 1 Table: public.users

Columns






	Name	Data type	Description / Attributes
	id	integer	<b>Default:</b> nextval('users_id_seq'::regclass)
	username	text	
	firstname	text	
	lastname	text	
	email	text	
	phone	text	
	passwd	text	

Unique keys



Columns	Name / Description
 id	users_pkey

## 1.2.1. 2 Table: public.nodes


Columns

	Name	Data type	Description / Attributes
	id	integer	<b>Default:</b> nextval('users_id_seq'::regclass)
	name	text	
	ip	text	
	des	text	
	status	Boolean	<b>Nullable Default:</b> false

Linked from

	Table	Join	Title / Name / Description
	public.action	<b>public.nodes.id =</b> public.action.node_id	<b>action_node_id_fkey</b>
	public.alarms	<b>public.nodes.id =</b> public.alarms.node_id	<b>alarms_node_id_fkey</b>






Unique keys

Columns	Name / Description
 id	nodes_pkey




## 1.2.1.3 Table: public.action


Columns

	Name	Data type	Description / Attributes
	id	integer	<b>Default:</b> nextval('users_id_seq'::regclass)
	node_id	integer	<b>Nullable</b> <b>References:</b> public.nodes
	alarm_type	text	
	action	text	
	date	date	<b>Default:</b> now()

Linked to







	Table	Join	Title / Name / Description
	public.nodes	<b>public.action.node_id = public.nodes.id</b>	<b>action_node_id_fkey</b>

Unique keys


	Columns	Name / Description
	id	action_pkey

## 1.2.1. 4 Table: public.alarms


### Columns

	Name	Data type	Description / Attributes
	id	integer	<b>Default:</b> nextval(alarms_id_seq::regclass)
	node_id	integer	<b>Nullable</b> <b>References:</b> public.nodes
	alarm_type	text	
	oid	text	
	des	text	
	status	boolean	<b>Nullable</b> <b>Default:</b> false


### Linked to

	Table	Join	Title / Name / Description
	public.nodes	<b>public.alarms.node_id</b> = <b>public.nodes.id</b>	<b>alarms_node_id_fkey</b>

### Unique keys

	Columns	Name / Description
	id	alarms_pkey


### Triggers

	Name	When	Description
	statustrigger	After Insert	

## 1.2.2 Functions

### 1.2.1. 1 Function: public. updatestatus

Input/output

	Name	Data type	Description
	Returns	trigger	

# CHAPTER

# TWO

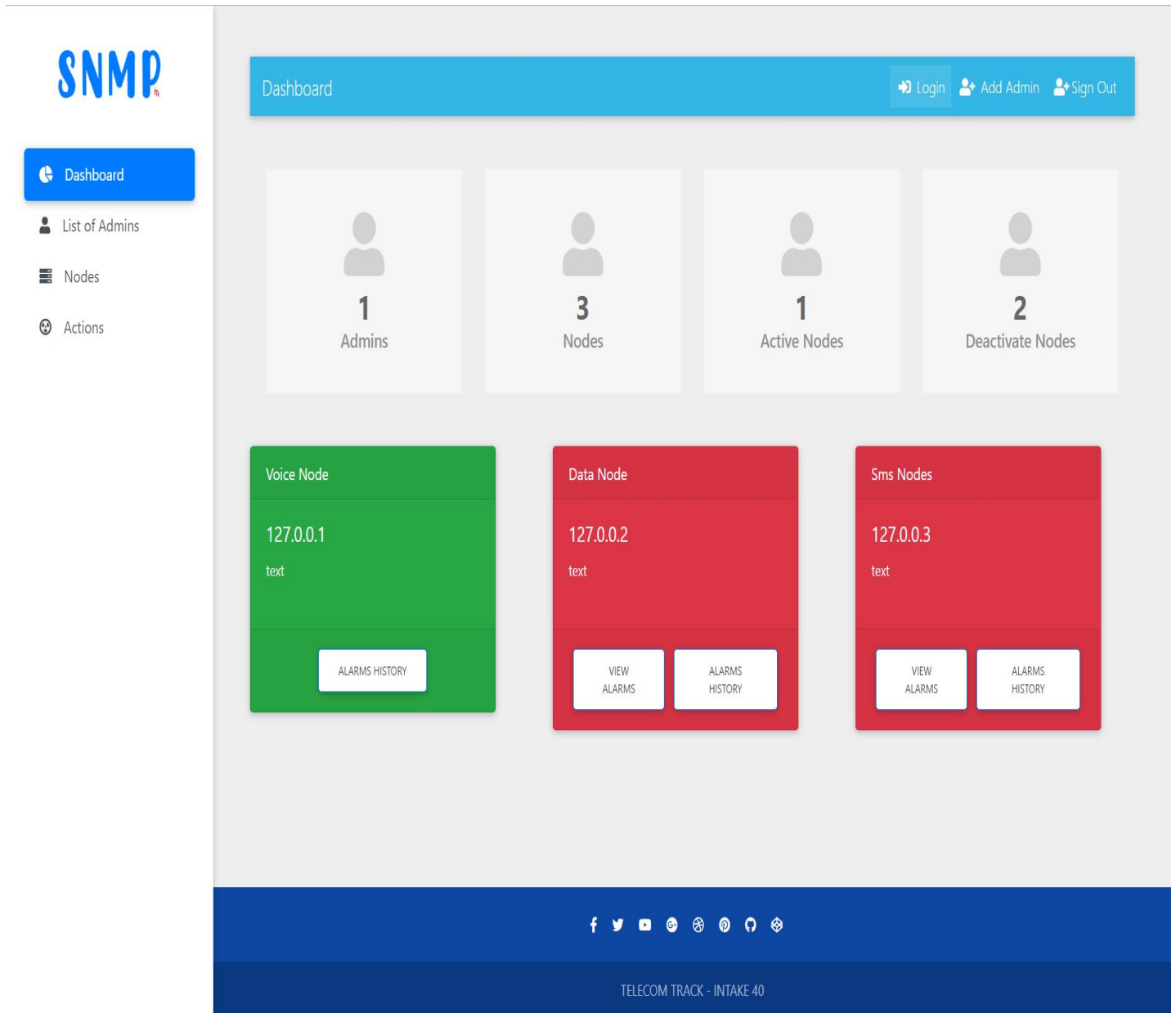
# WEB

# INTERFACE

SNMP PLATFORM

## CH02: Web Interface

### 2.0 Web Interface Screen Shot



## 2.1 Web Interface Features

- ❖ Register an account
- ❖ Login
- ❖ Counter for admins, nodes, Active Nodes, Deactivate Nodes
- ❖ Display the status of the node
- ❖ Check Node is Live or not
- ❖ Alarms History
- ❖ view Alarms
- ❖ List of Admins
- ❖ Add, Delete and Update Nodes
- ❖ Add, Delete and Update Actions
- ❖ Make Action
- ❖ Notification
- ❖ PostgreSQL cloud databases
- ❖ web hosting on cloud



# CHAPTER THREE SNMP CLIENT & SERVER

SNMP PLATFORM

## CH03: SNMP Client & Server

---

### 3.0 What is snmp?

SNMP stands for Simple Network Management Protocol, is a communication protocol that lets you monitor managed network devices including Routers, Switches, Servers, Printers and other devices that are IP enabled all through a single management system/software.

### 3.1 What does SNMP do?

- Monitor inbound and outbound Traffic flowing through the device
- Early Detection of faults within network devices along with Alerts/Notifications
- Analyzing data collected from devices over long periods of time to identify bottlenecks and performance issues
- Ability to remotely configure compatible devices
- Access and Control devices remotely that are connected via SNMP

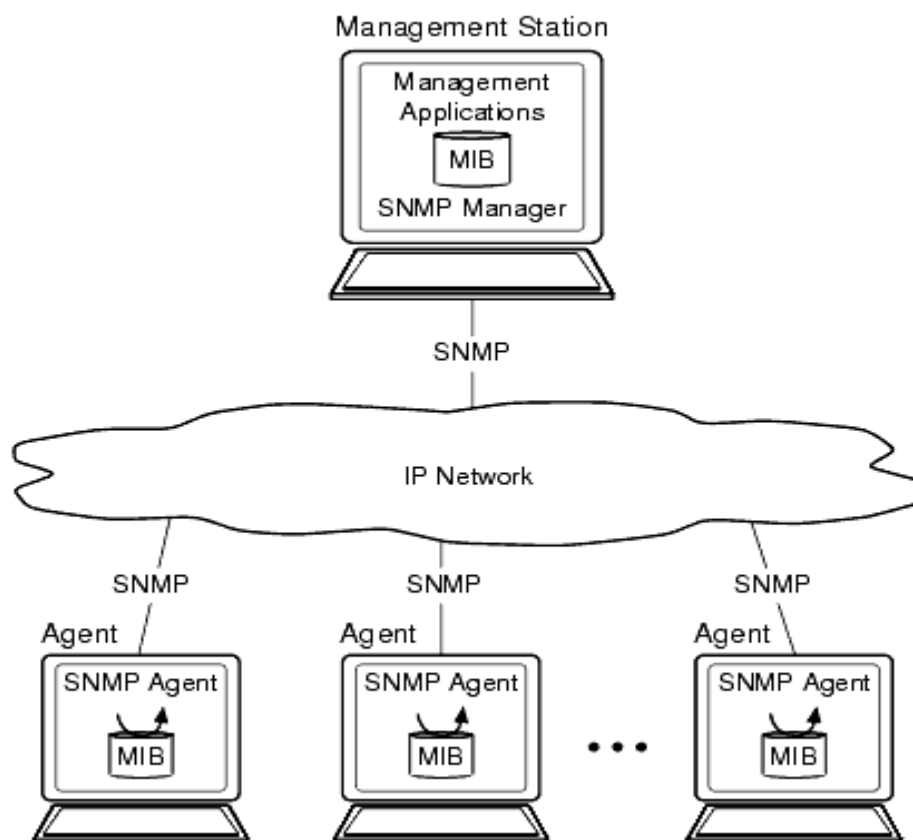
### 3.2 snmpTerminology

- Manager (the monitoring "client")
- Agent (running on the equipment/server)



## 3.3 What is MIB (management information base)?

- Each management station or agent in an SNMP-managed network maintains a local database of information relevant to network management, known as the management information base (MIB). The relationship between the management station, the agent, and the MIB is shown in the following figure.



- An SNMP-compliant MIB contains definitions and information about the properties of managed resources and the services that the agents support. The manageable features of resources, as defined in an SNMP-compliant MIB, are called *managed objects* or *management variables* (or just objects or variables). A management station gets and sets objects in the MIB, and an agent notifies the management station of significant but unsolicited events called *traps*. All message exchanges between the management station and its agents take place using the Simple Network Management Protocol (SNMP). The MIB at the management station contains network management information extracted from the MIBs of all the managed entities in the network.

## 3.4 OID

The first part of the Trap OID is the vendor's identification number of the network management system contained in the entity. In the preceding example, this number is the enterprise identification number. The last two digits of the Trap OID indicate products and the software respectively.

## 3.5 Snmp Operations

- Get
- Get next
- Get bulk
- Set
- Get response
- Trap
- Notification
- Inform
- Report

Snmp protocol use port 161 for query And port 162 to send a trap.

Most common operation used is

- Get

Query for a value (manager -> agent) some typical commands for querying:

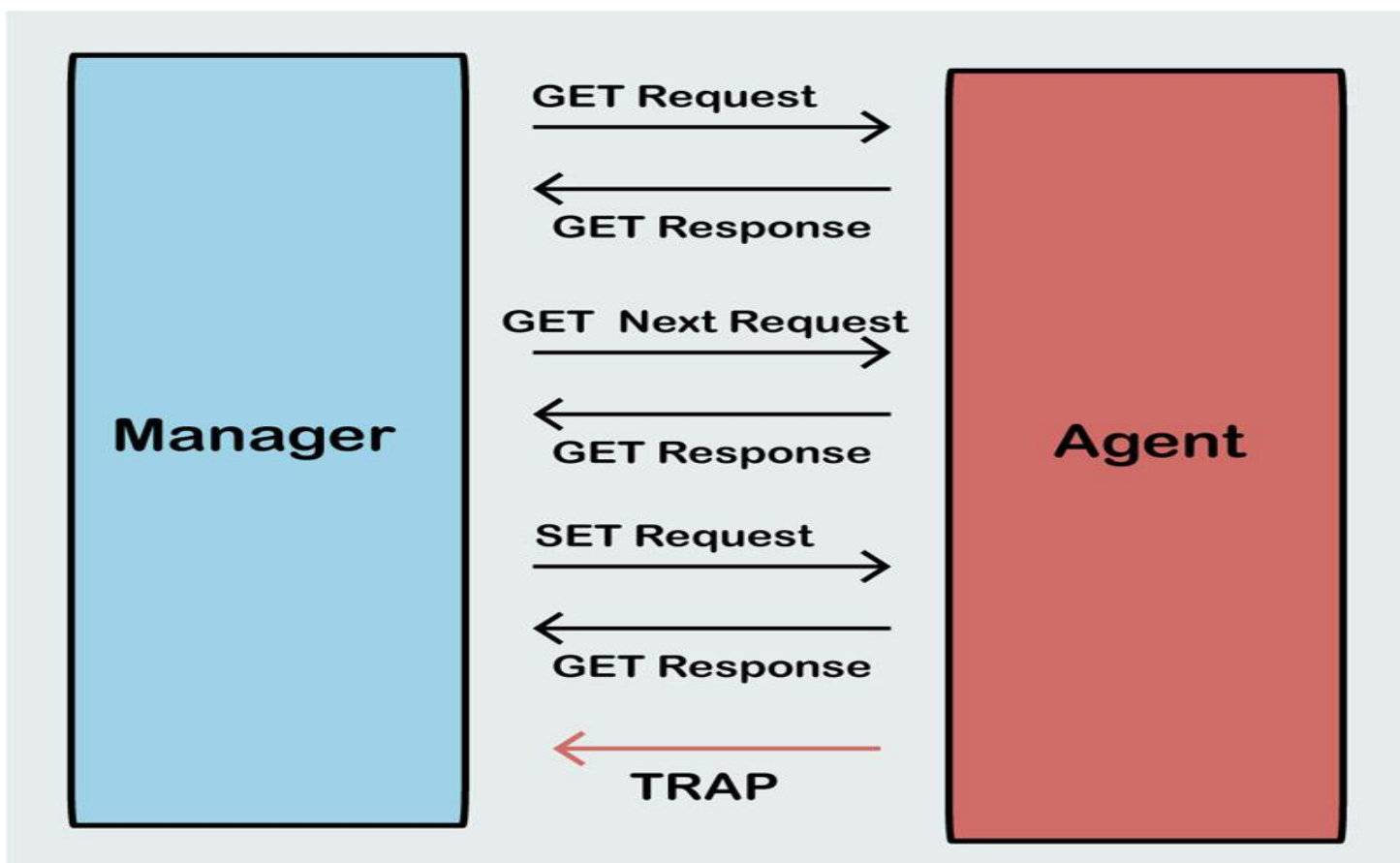
- snmpget
- snmpwalk
- snmpstatus
- snmptable

- GET-RESPONSE

Response to GET/SET, or error (agent -> manager)

- **3.6 Trap**

SNMP Traps are the most frequently used alert messages sent from a remote SNMP-enabled device (an agent) to a central collector, the “SNMP manager.” As mentioned earlier, the Trap messages are the main form of communication between an SNMP Agent and SNMP Manager. They are used to inform an SNMP manager when a significant event occurs at the Agent level. What makes the Trap unique from other messages is that they are triggered instantaneously by an agent, rather than waiting for a status request from the SNMP Manager.





# CHAPTER FOUR MAKE ACTION

SNMP PLATFORM

## CH04: Make Action

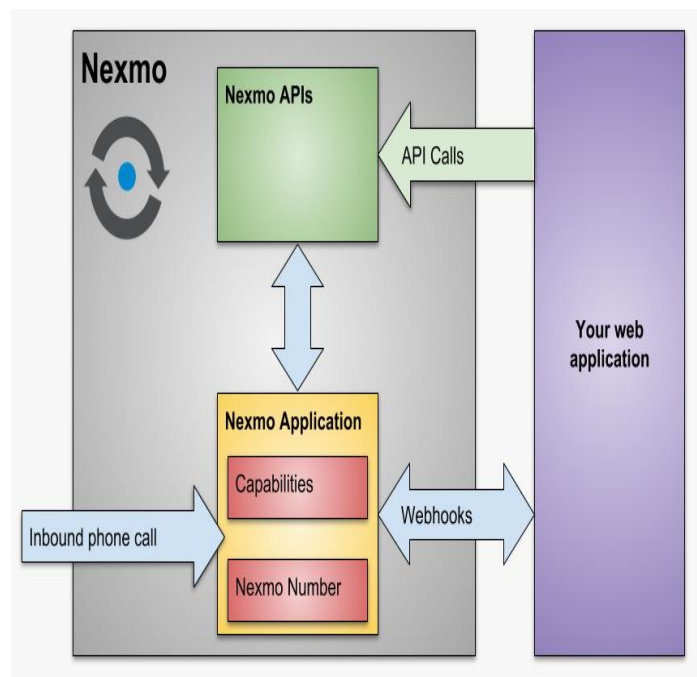
### • 4.0 Sending SMS using Nexmo

#### What is Vonage Communications APIs (formerly Nexmo)?

Nexmo, the Vonage API Platform, enables you to engage your customers in real time, anywhere in the world. Developers and businesses of all sizes use Nexmo APIs to embed contextual, programmable communications into their mobile apps, websites and business systems via text, voice, video, social media and messaging. With easy-to-use APIs, a global platform, and expert support, Nexmo enables companies to abstract the complexity of communications and instead focus on building innovative, delightful customer experiences. For developers, the Nexmo platform provides a comprehensive set of tools and resources to get started quickly and build applications easily. Managing your Nexmo account is just as easy. You can use the simple but powerful Nexmo Dashboard or manage your account directly through an API. From its inception as a disruptive startup, Nexmo has been on a mission to reinvent communication. We have pursued that mission by empowering developers and companies to innovate their customer experiences and scale globally without having to worry about the telecommunications details. In June of 2016, Nexmo was acquired by Vonage and became the Vonage API platform. Joining forces with a leading provider of business cloud communications further strengthened our ability to fulfill our bold mission.

A Vonage API application contains the security and configuration information you need to connect to Vonage endpoints and use the Vonage APIs.

Each Vonage application created can support multiple capabilities - for example you can create an Application that supports using the Voice, Messages and RTC APIs.



- **4.1 What is Nexmo?**

Nexmo connects your apps directly to carriers around the world. Integrate SMS and Voice messages using one simple API. Use Nexmo Verity to register users, verify transactions, and implement two-factor authentication.

Nexmo is a tool in the **Voice and SMS** category of a tech stack.

- **4.2 Who uses Nexmo?**

## COMPANIES

**43** companies reportedly use **Nexmo** in their tech stacks, including **Airbnb**, **N26**, and **Practo**.



## DEVELOPERS

**81** developers on StackShare have stated that they use **Nexmo**



Nexmo is loved by developers and built for business.

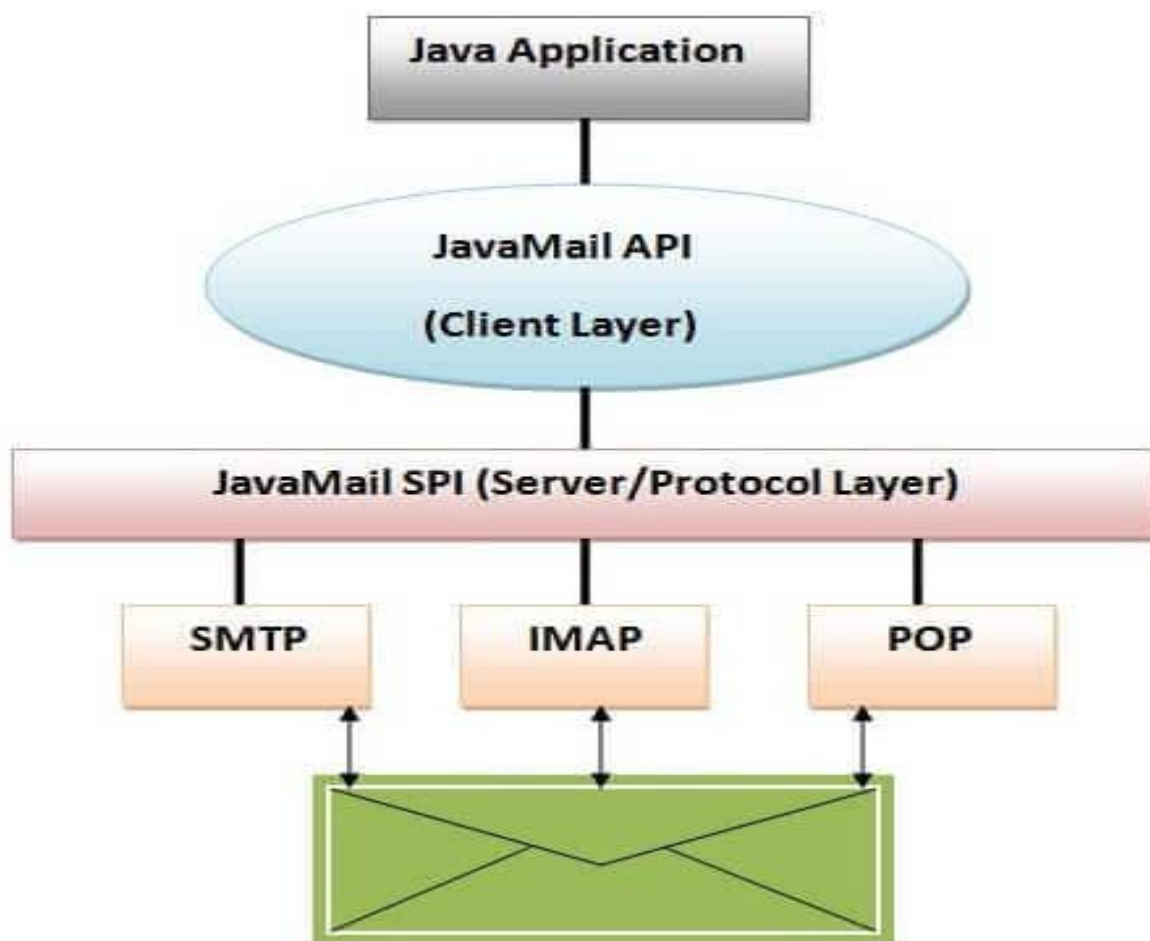
1. Nexmo Innovates Customer Engagement with Omni channel Interface
2. Nexmo: Loved by developers. Built for business.
3. Nexmo: Enhancing the Customer Experience with Communications APIs

- **4.3 Sending an Email using the JavaMail API.**

## Introduction

Sending emails is one of the common tasks in real life applications and that's why Java provides robust **JavaMail API** that we can use to send emails using SMTP server. JavaMail API supports both TLS and SSL authentication for sending emails.

The JavaMail API defines classes which represent the components of a mail system. JavaMail does not implement an email server, instead it allows you to access an email server using a Java API. In order to test the code presented in this tutorial you must have access to an email server. While the JavaMail API specification does not mandate support for specific protocols, JavaMail typically includes support for POP3, IMAP, and SMTP. This tutorial covers sending mail messages and only requires access to an SMTP server.





## Scenario

*Sending an email from within a Java application has many practical uses. A web application may allow visitors to create an account. Part of the account creation process could include sending an email with a unique activation code to verify that a valid email address was given at signup. Online shopping applications typically notify buyers of order or tracking status via email.*

## Hardware and Software Requirements

The following is a list of hardware and software requirements:

- Download and install Java JDK 7
- Download and install an IDE and Application Server such as NetBeans 7.1.2 with Java EE which includes GlassFish 3.1.2 (Java EE download bundle). During installation, be sure to check the box to install GlassFish. JUnit is an optional installation and not required for this tutorial.
- Access to an SMTP server
- Valid credentials (username and password) for the SMTP server

## Prerequisites

Before starting this tutorial, you should:

- Have access to an SMTP server. You must know the host name, port number, and security settings for your SMTP server. Web mail providers may offer SMTP access, view your email account settings or help to find further information. Be aware that your username is often your full email address and not just the name that comes before the @ symbol.
- A Java EE IDE and Application Server such as GlassFish or Oracle WebLogic Server. JavaMail can be [downloaded](#) and used as a library in a Java SE application but this tutorial assumes the use of a Java EE application server which would already include JavaMail.
- Have basic familiarity with Servlets and Session EJBs (helpful but not required)



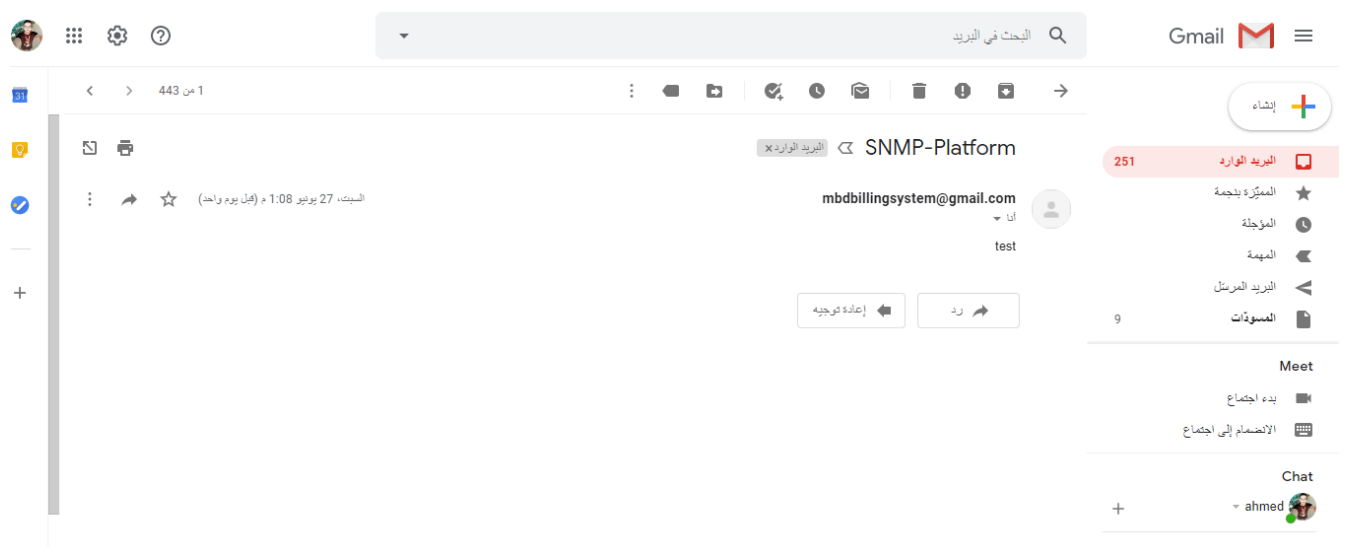
## To send emails using Java, you need three things:

- JavaMail API
- Java Activation Framework (JAF)
- Your SMTP server details

You may download the latest version of both JavaMail API and JAF from the official website of Java. After successfully downloading these two, extract them. Add **mail.jar**, **smtp.jar** and **activation.jar** file in your classpath to be eligible to send emails.

After adding these files, follow the below steps and write a java program to send email:

- Create a new session object by calling `getDefaultInstance()` method and passing properties as argument to get all of the important properties like hostname of the SMTP server etc.
- Create a `MimeMessage` object by passing the session object created in previous step.
- The final step is to send email using the `javax.mail.Transport`



## ● 4.4 Run a Shell Script in Java.

### How do I run a .sh file in Java?

Using Runtime.getRuntime().exec to **execute shell script**

After **executing** this **Java** program with the given **shell script**, if you verify at the location where your **Java** program is you will see a directory test created and a **file** SAMPLE with in that directory. Runtime.getRuntime().exec method is used to **run** the command.

In the end, these solutions are implemented with two ways:

- 1- Either automatically without the intervention of the admin
- 2- With the intervention of the admin through Java web application.