**TELECOM PORTAL**

**ASTERISK – JAVA**

**ASTERISK**

Asterisk is an open source software PBX, created by [Digium](http://www.voip-info.org/wiki/view/Digium" \o "Digium), the Asterisk source code and low cost telephony hardware that works with Asterisk. Asterisk runs on Linux and other Unix platforms **with OR without** hardware that connects server to the traditional global telephony network, the [PSTN](http://www.voip-info.org/wiki/view/PSTN).

**ASTERISK MANAGER INTERFACE (AMI)**

The Asterisk Manager Interface (AMI) allows a client program to connect to an Asterisk instance and issue commands or read events over a TCP/IP stream. Integrators will find this particularly useful when trying to track the state of a telephony client inside Asterisk, and directing that client based on custom (and possibly dynamic) rules.

**ASTERISK GATEWAY INTERFACE (AGI)**

When calls come into the Asterisk server, the dial plan rules process the call and determine where to route it. To launch an AGI program and hand off call processing the AGI program, you will need to use the Asterisk dial plan command AGI. Below is an extremely simple example dial plan which passes all calls to an AGI script for processing.

The AGI is a middle man, lying somewhere between dial plan and the AMI in terms of functionality. Then it is usable only for incoming calls, and is thus no good for purely outbound telephony development

**TYPES**

* STANDARD AGI: Simple and most AGI. Standard AGI scripts run on the local PBX and communicate with Asterisk through socket.
* DEAD AGI: Continues working on after channel hang up.
* FAST AGI: For large application
* EAGI: Give a way to access the audio channel directly for the calls

**ELASTIX**

Elastix is an application software that integrates with Asterisk-based PBXs into a single, easy-to-use interface. It also adds its own set of utilities and allows for the creation of third party modules.

**SOFTWARE REQUIREMENT**

* Java Jdk 8
* Asterisk Java Jar file

**ASTERISK JAVA**

The [Asterisk-Java](http://asterisk-java.org/) package(jar) consists of a set of Java classes that allow you to easily build Java applications that interact with an Asterisk PBX Server. [Its](http://asterisk-java.org/) supports both interfaces that Asterisk provides for this scenario: The [FastAGI](http://www.voip-info.org/wiki/view/Asterisk+FastAGI" \o "Asterisk FastAGI) protocol and the [Manager API](http://www.voip-info.org/wiki/view/Asterisk+manager+API).

The FastAGI implementation supports all [commands](http://asterisk-java.org/development/apidocs/org/asteriskjava/fastagi/command/package-summary.html) currently available from Asterisk.

The Manager API implementation supports receiving events from the Asterisk server (e.g. call progress, registered peers, channel state, channel status, live calls) and sending actions to Asterisk (e.g. originate call, agent login/logoff, transfer call, Hang up call , Park call start/stop voice recording).

**TASK DETAILS**

1. Live incoming & outgoing calls should be triggered.
2. Live Conference calls should be triggered.
3. Live Parking calls should be triggered.
4. Live Queue calls should be triggered.
5. Live call transfer should be done through our dashboard.
6. Live Call can be disconnected from our dashboard.
7. Call Origination should be done through our dashboard.
8. Live Call should be hold from our dashboard.

**TASK 1:**

**LIVE INCOMING & OUTGOING CALLS SHOULD BE TRIGGERED**

Using asterisk manager interface, I have used following functions (org.asteriskjava.manager), - LiveCallFile.java

* **ManagerConnection** for connecting AMI with program,
* **CommandAction** for execute the asterisk command,
* **CommandResponse** for getting response from command action,
* **SendAction** for send the Action to Asterisk ManagerConnection.
* **BufferedWriter** to write command response into output.txt file.

**TASK 2:**

**LIVE CONFERENCE CALLS SHOULD BE TRIGGERED**

Using asterisk manager interface, I have used following functions (org.asteriskjava.manager), - MeetmeRoom.java

* **AsteriskServer** for connecting AMI with program,
* **CommandAction** for execute the asterisk command,
* **CommandResponse** for getting response from command action,
* **SendAction** for send the Action to Asterisk ManagerConnection.
* **BufferedWriter** to write command response into MeetmeRoom.txt file.

**TASK 3:**

**LIVE PARKING CALLS SHOULD BE TRIGGERED**

Using asterisk manager interface, I have used following functions (org.asteriskjava.manager), - PakingCalls.java

* **ManagerConnection** for connecting AMI with program,
* **CommandAction** for execute the asterisk command,
* **CommandResponse** for getting response from command action,
* **SendAction** for send the Action to Asterisk ManagerConnection.
* **BufferedWriter** to write command response into ParkingCalls.txt file.

**TASK 4:**

**LIVE QUEUE CALLS SHOULD BE TRIGGERED**

Using asterisk manager interface, I have used following functions (org.asteriskjava.manager), - Queue.java

* **ManagerConnection** for connecting AMI with program,
* **CommandAction** for execute the asterisk command,
* **CommandResponse** for getting response from command action,
* **SendAction** for send the Action to Asterisk ManagerConnection.
* **BufferedWriter** to write command response into Queue.txt file.

**TASK 5:**

**LIVE CALL TRANSFER SHOULD BE DONE THROUGH OUR DASHBOARD**

Using asterisk manager interface, I have used following functions (org.asteriskjava.manager), - CallTransferBoth.java

* **ManagerConnection** for connecting AMI with program,
* **CommandAction** for execute the asterisk command,
* **CommandResponse** for getting response from command action,
* **ManagerResponse** for getting response from manager action
* **SendAction** for send the Action to Asterisk ManagerConnection.
* **RedirectAction** to redirect a live channel to some other extension.

**TASK 6:**

**LIVE CALL CAN BE DISCONNECTED FROM OUR DASHBOARD**

Using asterisk manager interface, I have used following functions (org.asteriskjava.manager), - HangUp1.java

* **ManagerConnection** for connecting AMI with program,
* **CommandAction** for execute the asterisk command,
* **CommandResponse** for getting response from command action,
* **ManagerResponse** for getting response from manager action
* **SendAction** for send the Action to Asterisk ManagerConnection.

**TASK 7:**

**CALL ORIGINATE SHOULD BE DONE THROUGH OUR DASHBOARD**

Using asterisk manager interface, I have used following functions (org.asteriskjava.manager), - CallOriginate.java

* **ManagerConnection** for connecting AMI with program,
* **CommandAction** for execute the asterisk command,
* **CommandResponse** for getting response from command action,
* **ManagerResponse** for getting response from manager action
* **SendAction** for send the Action to Asterisk Manager Connection.
* **OriginateAction** for originate a new call from extension to some other number.

**TASK 8:**

**LIVE CALL SHOULD BE HOLD FROM OUR DASHBOARD**

Using asterisk manager interface, I have used following functions (org.asteriskjava.manager), - HoldCall.java

* **ManagerConnection** for connecting AMI with program,
* **CommandAction** for execute the asterisk command,
* **CommandResponse** for getting response from command action,
* **ManagerResponse** for getting response from manager action
* **SendAction** for send the Action to Asterisk Manager Connection.
* **ParkingAction** for hold the channel to the same channel for some time.

**FILE DIRECTORY, FILE NAME AND EXECUTION METHOD,**

**For Live calls,**

* Live calls( output.txt) - LiveCallFile.java
* Conference calls(MeetmeRoom.txt) –MeetmeRoom.java
* Parking calls(Hold- ParkingCalls.txt) – ParkingCalls.java
* Queue(Waiting- Queue.txt) – Queue.java
  + **Directory path**              **:**    /var/www/html/telecom/incomefile/LiveJava
  + **File name(Shell Script)  :**Worker.sh
  + **Execution method          :**sh Worker.sh

**For Other Actions:**

* Call Originate(Dial),
  + **Directory path**              **:**    /var/www/html/telecom/incomefile/Actions
  + **File name(Java File)      :**OriginateCall.java
  + **Execution method          :**Java -cp "asterisk-java.jar" OriginateCall <Source Extension num> <Destination Extension num> <SIP/Dahdi>
* Call Disconnect(HangUp),
  + **Directory path**              **:**    /var/www/html/telecom/incomefile/Actions
  + **File name(Java File)      :**HangUp1.java
  + **Execution method          :**Java -cp "asterisk-java.jar" Hangup1 <Extension number to hangup>
* Call Hold,
  + **Directory path**              **:**    /var/www/html/telecom/incomefile/Actions
  + **File name(Java File)      :**HoldCall.java
  + **Execution method          :**Java -cp "asterisk-java.jar" HoldCall <Extension number to Hold> <Channel to hold>
* Call Transfer(Both Blind and attended calls),

             Types,

* + Blind transfer
  + Attended Transfer
  + Transfer to queue
  + Transfer to conference
* **Directory path**              **:**    /var/www/html/telecom/incomefile/Actions
* **File name(Java File)      :** CallTransferBoth.java
* **Execution method          :**Java -cp "asterisk-java.jar" CallTransferBoth <Source Extension num> <Destination Extension num>
* Peer Info,
  + **Directory path                :** /var/www/html/telecom/incomefile/Actions
  + **File name(Java File)      :**PeerInfo.java
  + **File name(txt File)         :**PeerInfo.txt
  + **Execution method          :**Java -cp "asterisk-java.jar" PeerInfo <Peer Number>