Sip Connectors

Table of Contents

Restcomm SIP Servlets comes with default settings that are designed to get your system up and running without the need to know about all the detailed configurations. That said, there are situations in which you might like to fine-tune your setttings to adapt it to your needs. That is what the following section will help you achieve. You will get a better understand of SIP connectors and how to make them work for you.

Configuring SIP Connectors and Bindings

There are two important configuration files that you might need to modifying depending on your system needs. The standalone-sip.xml file in Restcomm SIP Servlets for JBoss AS7 and the server.xml file in Restcomm SIP Servlets for Tomcat. The extracts below will give you a snapshot of default configurations.

For JBoss

Changing the ports and other configuration for the SIP connector can be done in the standalone-sip.xml file. Below is an extract :

Example 1. Adding a SIP Connector to $JBOSS_HOME/standalone/configuration/standalone-sip.xml

```
 <socket-binding-group name="standard-sockets" default-interface="public" port-
offset="${jboss.socket.binding.port-offset:0}">
        <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:9999}"/>
        <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:9990}"/>
        <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:9443}"/>
        <socket-binding name="ajp" port="8009"/>
        <socket-binding name="http" port="8080"/>
        <socket-binding name="https" port="8443"/>
        <socket-binding name="sip-udp" port="5080"/>
        <socket-binding name="sip-tcp" port="5080"/>
        <socket-binding name="sip-tls" port="5081"/>
    <socket-binding name="sip-ws" port="5082"/>
        <socket-binding name="osgi-http" interface="management" port="8090"/>
        <socket-binding name="remoting" port="4447"/>
        <socket-binding name="txn-recovery-environment" port="4712"/>
        <socket-binding name="txn-status-manager" port="4713"/>
        <outbound-socket-binding name="mail-smtp">
            <remote-destination host="localhost" port="25"/>
        </outbound-socket-binding>
    </socket-binding-group>
```

If you need to add a connector for the same protocol, a new socket-binding should be created. A

naming convention should be followed for the name attribute of the new socket-binding. The convention is <name>-sip-<protocol>. By example,

```
    <socket-binding name="second-sip-udp" port="5080"/>
```

SIP <connector> Attributes

port (defined at the socket-binding element)

The port number on which the container will be able to receive SIP messages.

protocol

Specifies the connector is a SIP Connector and not an HTTP Connector. There is no need to change this property.

signalingTransport (use socket-binding element)

Specifies the transport on which the container will be able to receive SIP messages. Supported Values are "udp", "tcp", "tls" and "ws".

use-stun

Enables Session Traversal Utilities for NAT (STUN) support for this Connector. The attribute defaults to "false". If set to "true", ensure that the `ipAddress` attribute is *not* set to `127.0.0.1`. Refer to Restcomm SIP Servlets [_mssstun__stun] for more information about STUN.

stun-server-address

Specifies the STUN server address used to discover the public IP address of the SIP Connector. This attribute is only required if the `useStun` attribute is set to "true". Refer to Restcomm SIP Servlets [_mssstun__stun] for more information about STUN and public STUN servers.

stun-server-port

Specifies the STUN server port of the STUN server used in the `stunServerAddress` attribute. You should rarely need to change this attribute; also, it is only needed if the `useStun` attribute is set to "true". Refer to Restcomm SIP Servlets [_mssstun__stun] for more information about STUN.

use-static-address

Specifies whether the settings in staticServerAddress and staticServerPort are activated. The default value is "false" (deactivated).

static-server-address

Specifies what load-balancer server address is inserted in Contact/Via headers for server-created requests. This parameter is useful for cluster configurations where requests should be bound to a load-balancer address, rather than a specific node address.

static-server-port

Specifies the port of the load-balancer specified in staticServerAddress. This parameter is useful in cluster configurations where requests should be bound to a load-balancer address rather than a specific node address.

http-follow-sip

Makes the application server aware of how the SIP Load Balancers assign request affinity, and stores this information in the application session.

For Tomcat

Changing the ports and other configuration for the SIP connector can be done in the server.xml file. Below is an extract.

Example 2. Adding a SIP Connector to $CATALINA_HOME/conf/server.xml

```
  <Connector port="5080"
ipAddress="127.0.0.1"
 protocol="org.mobicents.servlet.sip.startup.SipProtocolHandler"
 signalingTransport="udp"
 useStun="false"
 stunServerAddress="stun01.sipphone.com"
 stunServerPort="3478"
 staticServerAddress="122.122.122.122"
 staticServerPort="44"
 useStaticAddress="true"
 httpFollowsSip="false"/>
```

SIP <connector> Attributes

port

The port number on which the container will be able to receive SIP messages.

ipAddress

The IP address at which the container will be able to receive SIP messages. The container can be configured to listen to all available IP addresses by setting `ipAddress` to `0.0.0.0 <sipPathName>`.

protocol

Specifies the connector is a SIP Connector and not an HTTP Connector. There is no need to change this property.

signalingTransport

Specifies the transport on which the container will be able to receive SIP messages. For example, "udp".

useStun

Enables Session Traversal Utilities for NAT (STUN) support for this Connector. The attribute defaults to "false". If set to "true", ensure that the `ipAddress` attribute is *not* set to `127.0.0.1`. Refer to Restcomm SIP Servlets [_mssstun__stun] for more information about STUN.

stunServerAddress

Specifies the STUN server address used to discover the public IP address of the SIP Connector. This attribute is only required if the `useStun` attribute is set to "true". Refer to Restcomm SIP

Servlets [_mssstun__stun] for more information about STUN and public STUN servers.

stunServerPort

Specifies the STUN server port of the STUN server used in the `stunServerAddress` attribute. You should rarely need to change this attribute; also, it is only needed if the `useStun` attribute is set to "true". Refer to Restcomm SIP Servlets [_mssstun__stun] for more information about STUN.

useStaticAddress

Specifies whether the settings in staticServerAddress and staticServerPort are activated. The default value is "false" (deactivated).

staticServerAddress

Specifies what load-balancer server address is inserted in Contact/Via headers for server-created requests. This parameter is useful for cluster configurations where requests should be bound to a load-balancer address, rather than a specific node address.

staticServerPort

Specifies the port of the load-balancer specified in staticServerAddress. This parameter is useful in cluster configurations where requests should be bound to a load-balancer address rather than a specific node address.

httpFollowsSip

Makes the application server aware of how the SIP Load Balancers assign request affinity, and stores this information in the application session.

A comprehensive list of implementing classes for the SIP Stack is available from the Class SipStackImpl page on nist.gov.

Application Routing and Service Configuration

The application router is called by the container to select a SIP Servlet application to service an initial request. It embodies the logic used to choose which applications to invoke. An application router is required for a container to function, but it is a separate logical entity from the container.

The application router is responsible for application selection and must not implement application logic. For example, the application router cannot modify a request or send a response.

For more information about the application router, refer to the following sections of the JSR 289 specification: Application Router Packaging and Deployment, Application Selection Process, and Appendix C.

.

See the example chapters for more information about the Application Router Configuration for SIP Restcomm SIP Servlets for JBoss AS7

[_sfss_services_for_sip_servlets]

In order to configure the application router for Tomcat, you should edit the `Service` element in the container's *server.xml* configuration file

Example 3. Configuring the Service Element in the Container's server.xml

```
 <Service name="Sip-Servlets"
 className="org.mobicents.servlet.sip.startup.SipStandardService"
 sipApplicationDispatcherClassName="org.mobicents.servlet.sip.core.SipApplicationD
ispatcherImpl"
 usePrettyEncoding="false"
 additionalParameterableHeaders="Header1,Header2"
 bypassResponseExecutor="false"
 bypassRequestExecutor="false"
 baseTimerInterval="500"
 t2Interval="4000"
 t4Interval="5000"
 timerDInterval="32000"
 dispatcherThreadPoolSize="4"
 darConfigurationFileLocation="file:///home/user/workspaces/sip-servlets/
 sip-servlets-examples/reinvite-demo/reinvite-dar.properties"
 sipStackPropertiesFile="conf/mss-sip-stack.properties"
 dialogPendingRequestChecking="false"
 callIdMaxLength="32"
 tagHashMaxLength="10"
 canceledTimerTasksPurgePeriod="1">
```

For Restcomm SIP Servlets for JBoss AS7 this is located in standalone-sip.xml file :

Example 4. Configuring the Mobicents SubSystem Element in the Container's standalone.xml

```
  <subsystem xmlns="urn:org.mobicents:sip-servlets-as7:1.0" application-
router="dars/mobicents-dar.properties" stack-properties="mss-sip-stack.properties"
path-name="gov.nist" app-dispatcher-
class="org.mobicents.servlet.sip.core.SipApplicationDispatcherImpl" concurrency-
control-mode="SipApplicationSession" congestion-control-interval="-1">
            <connector name="sip-udp" protocol="SIP/2.0" scheme="sip" socket-
binding="sip-udp"/>
            <connector name="sip-tcp" protocol="SIP/2.0" scheme="sip" socket-
binding="sip-tcp"/>
            <connector name="sip-tls" protocol="SIP/2.0" scheme="sip" socket-
binding="sip-tls"/>
        </subsystem>
```

SIP Service element attributes

className

This attribute specifies that the servlet container is a *converged* (i.e. SIP + HTTP) servlet container.

sipApplicationDispatcherClassName (Tomcat) - app-dispatcher-class (JBoss/EAP)

This attribute specifies the class name of the `org.mobicents.servlet.sip.core.SipApplicationDispatcher` implementation to use. The routing algorithm and application selection process is performed in that class.

darConfigurationFileLocation (Tomcat) - application-router (JBoss/EAP)

The default application router file location. This is used by the default application router to determine the application selection logic. Refer to Appendix C of the JSR 289 specification for more details.

sipStackPropertiesFile (Tomcat) - stack-properties (JBoss/EAP)

Specifies the location of the file containing key value pairs corresponding to the SIP Stack configuration properties. This attribute is used to further tune the JAIN SIP Stack. If this property is omitted, the following default values are assumed:

usePrettyEncoding (Tomcat) - use-pretty-encoding (JBoss/EAP)

Allows Via, Route, and RecordRoute header field information to be split into multiple lines, rather than each header field being separating with a comma. The attribute defaults to "true". Leaving this attribute at the default setting may assist in debugging non-RFC3261 compliant SIP servers.

additionalParameterableHeaders (Tomcat) - additional-parameterable-headers (JBoss/EAP)

Comma separated list of header names that are treated as parameterable by the container. The specified headers are classed as valid, in addition to the standard parameterable headers defined in the Sip Servlets 1.1 Specification.

baseTimerInterval (Tomcat) - base-timer-interval (JBoss/EAP)

Specifies the `T1` Base Timer Interval, which allows the SIP Servlets container to adjust its timers depending on network conditions. The default interval is 500 (milliseconds).

t2Interval (Tomcat) - t2-interval (JBoss/EAP)

Specifies the `T2` Interval, which allows the SIP Servlets container to adjust its timers depending on network conditions. The default interval is 4000 (milliseconds).

t4Interval (Tomcat) - t4-interval (JBoss/EAP)

Specifies the `T4` Interval, which allows the SIP Servlets container to adjust its timers depending on network conditions. The default interval is 5000 (milliseconds).

timerDInterval (Tomcat) - timerD-interval (JBoss/EAP)

Specifies the `Timer D` Interval, which allows the SIP Servlets container to adjust its timers depending on network conditions. The default interval is 32000 (milliseconds).

dialogPendingRequestChecking (Tomcat) - dialog-pending-request-checking (JBoss/EAP)

This property enables and disables error checking when SIP transactions overlap. If within a single dialog an INVITE request arrives while there is antoher transaction proceeding, the container will send a 491 error response. The default value is false.

callIdMaxLength (Tomcat) - call-id-max-length (JBoss/EAP)

This property allows to shorten the size of Call-ID Header. This is useful when integrating with Lync (which has a limit of 32 in size) or older SIP Servers

tagHashMaxLength (Tomcat) - tag-hash-max-length (JBoss/EAP)

This property allows to shorten the size of tags in From and To Header. This is useful when integrating with Lync (which has a limit of 10 in size) or older SIP Servers

dnsServerLocatorClass (Tomcat) - dns-server-locator-class (JBoss/EAP)

Specifies the `org.mobicents.ext.javax.sip.dns.DNSServerLocator` implementation class that will be used by the container to perform DNS lookups compliant with RFC 3263 : Locating SIP Servers and E.164 NUmber Mapping. The default class used by the container is `org.mobicents.ext.javax.sip.dns.DefaultDNSServerLocator`, but any class implementing the `org.mobicents.ext.javax.sip.dns.DNSServerLocator` interface. To disable DNS lookups, this attribute should be left empty.

dnsResolverClass (Tomcat) - dns-resolver-class (JBoss/EAP)

Specifies the `org.mobicents.javax.servlet.sip.dns.DNSResolver` implementation class that will be used by the container to perform DNS lookups compliant with RFC 3263 : Locating SIP Servers and E.164 NUmber Mapping. The default class used by the container is `org.mobicents.servlet.sip.dns.MobicentsDNSResolver`, but any class implementing the `org.mobicents.servlet.sip.dns.DNSResolver` interface. To disable DNS lookups, this attribute should be left empty.

addressResolverClass (Tomcat) - address-resolver-class (JBoss/EAP)

Specifies the `gov.nist.core.net.AddressResolver` implementation class that will be used by the container to perform DNS lookups. The default class used by the container is

`org.mobicents.servlet.sip.core.DNSAddressResolver`, but any class implementing the `gov.nist.core.net.AddressResolver` NIST SIP Stack interface and having a constructor with a `org.mobicents.servlet.sip.core.SipApplicationDispatcher` parameter can be used. To disable DNS lookups, this attribute should be left empty.

canceledTimerTasksPurgePeriod (Tomcat) - canceled-timer-tasks-purge-period (JBoss/EAP)

Defines a period to due a purge in the container timer schedulers. The purge may prevent excessive memory usage for apps that cancel most of the timers it sets.

SIP Servlets Server Logging

Logging is an important part of working with Restcomm SIP Servlets. There are a few files that you need to be familiar with in order to successfully troubleshoot and adapt Restcomm SIP Servlets server monitoring and logging to your environment.

Logging Files for Restcomm SIP Servlets for JBoss AS7

$JBOSS/standalone/configuration/logging.properties

$JBOSS/standalone/configuration/mss-sip-stack.properties

$JBOSS/standalone/configuration/standalone-sip.xml

Example 5. Setting the log file name in $JBOSS/standalone/configuration/standalone-sip.xml

```
 </formatter>
<file relative-to="jboss.server.log.dir" path="server.log"/>
<suffix value=".yyyy-MM-dd"/>
<append value="true"/>
```

The configuration above produces SIP logs that can be found in the $JBOSS_HOME/standalone/log directory. Below is an extract of the log files.

```
                      server.log.2012-08-14  server.log.2012-08-24
server.log            server.log.2012-08-16  server.log.2012-08-25
server.log.2012-08-07 server.log.2012-08-21  server.log.2012-08-26
server.log.2012-08-13 server.log.2012-08-22
```

Logging Files for Restcomm SIP Servlets for Tomcat

If you are working with Tomcat, the log configuration files are located in the $CATALINA_HOME/conf/ directory. The log4j configuration file is located in $CATALINA_HOME/lib/ directory

$CATALINA_HOME/conf/logging.properties

$CATALINA_HOME/conf/mss-sip-stack.properties

$CATALINA_HOME/conf/server.xml

$CATALINA_HOME/lib/log4j.xml

Example 6. Setting the log file name $CATALINA_HOME/conf/server.xml

```
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
               prefix="localhost_access_log." suffix=".txt"
               pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" resolveHosts="false"/>
```

Example 7. Configuring the log file name $CATALINA_HOME/lib/log4j.xml

```
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">
  <appender name="rolling-file" class="org.apache.log4j.RollingFileAppender">
    <param name="file" value="${catalina.home}/logs/sip-server.log"/>
    <param name="MaxFileSize" value="1000KB"/>
```

The result of the extracted configuration above that is taken from the log4j.xml file and can be found in the $CATALINA_HOME/logs directory.

JAIN-SIP Stack Logging

There are two separate levels of logging:

- Logging at the container level, which can be configured using the *log4j.xml* or *standalone-sip.xml* configuration file seen above
- Logging of the JAIN SIP stack, which is configured through the container logging and the SIP stack properties themselves

You can setup the logging so that the JAIN SIP Stack will log into the container logs.

To use LOG4J in JAIN SIP Stack in Tomcat, you need to define a category in *CATALINE_HOME/lib/jboss-log4j.xml* and set it to `DEBUG`.

Example 8. Configuring the JAIN SIP Stack to log into the Tomcat Container's logs

```
                    <category name="gov.nist">
                    <priority value="DEBUG"/>
                    </category>
```

To use LOG4J in JAIN SIP Stack in JBoss, you need to define a logger in *JBOSS_HOME/standalone/configuration/standalone-sip.xml* and set it to `DEBUG`.

Example 9. Configuring the JAIN SIP Stack to log into the JBoss Container's logs

```
                    <logger category="gov.nist">
                    <level name="DEBUG"/>
                </logger>
```

For this category to be used in Restcomm SIP Servlets, you need to specify it in *JBOSS_HOME/standalone/configuration/mss-sip-stack.properties* or *CATALINE_HOME/conf/mss-sip-stack.properties*, add the `gov.nist.javax.sip.LOG4J_LOGGER_NAME=gov.nist` property, and set the `gov.nist.javax.sip.TRACE_LEVEL=LOG4J` property.