RestComm User Guide

The Guide to the RestComm

Jean Deruelle <jean.deruelle@gmail.com>
Thomas Quintana <quintana.thomas@gmail.com>
Charles Roufay <charles.roufay@telestax.com>
George Vagenas <gvagenas@telestax.com>

RestComm User Guide: The Guide to the RestComm

by Jean Deruelle, Thomas Quintana, Charles Roufay, and George Vagenas

Abstract

RestComm is a carrier-grade open source platform that provides developers the tools to integrate fax, voice, and SMS functionality in to their own applications with ease. RestComm is designed to have 100% compatibility with Twilio's APIs allowing easy porting between platforms. Furthermore, the RestComm platform is built on top of the industry leading Mobicents Sip Servlet Container and Mobicents Media Server providing the robustness and performance these platforms are already known to deliver.

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the Liberation Fonts [https://fedorahosted.org/liberation-fonts/] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight key caps and key-combinations. For example:

> To see the contents of the file `my_next_bestselling_novel` in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a key cap, all presented in Mono-spaced Bold and all distinguishable thanks to context.

Key-combinations can be distinguished from key caps by the hyphen connecting each part of a key-combination. For example:

> Press **Enter** to execute the command.
>
> Press **Ctrl**+**Alt**+**F1** to switch to the first virtual terminal. Press **Ctrl**+**Alt**+**F7** to return to your X-Windows session.

The first sentence highlights the particular key cap to press. The second highlights two sets of three key caps, each set pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in `Mono-spaced Bold`. For example:

> File-related classes include `filesystem` for file systems, `file` for files, and `dir` for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialogue box text; labelled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

> Choose System > Preferences > Mouse from the main menu bar to launch Mouse Preferences . In the Buttons tab, click the Left-handed mouse check box and click Close to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a gedit file, choose Applications > Accessories > Character Map from the main menu bar. Next, choose Search > Find… from the Character Map menu bar, type the name of the character in the Search field and click Next . The character you sought will be highlighted in the Character Table . Double-click this highlighted character to place it in the Text to copy field and then click the Copy button. Now switch back to your document and choose Edit > Paste from the gedit menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in Proportional Bold and all distinguishable by context.

Note the > shorthand used to indicate traversal through a menu and its sub-menus. This is to avoid the difficult-to-follow 'Select Mouse from the Preferences sub-menu in the System menu of the main menu bar' approach.

`Mono-spaced Bold Italic` or *`Proportional Bold Italic`*

Whether Mono-spaced Bold or Proportional Bold, the addition of Italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **ssh *username* @ *domain.name*** at a shell prompt. If the remote machine is `example.com` and your username on that machine is john, type **ssh john@example.com** .
>
> The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the `/home` file system, the command is **mount -o remount /home** .
>
> To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release*** .

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> When the Apache HTTP Server accepts requests, it dispatches child processes or threads to handle them. This group of child processes or threads is known as a *server-pool* . Under Apache HTTP Server 2.0, the responsibility for creating and maintaining these server-pools has been abstracted to a group of modules called *Multi-Processing Modules* (*MPMs*). Unlike other modules, only one module from the MPM group can be loaded by the Apache HTTP Server.

1.2. Pull-quote Conventions

Two, commonly multi-line, data types are set off visually from the surrounding text.

Output sent to a terminal is set in `Mono-spaced Roman` and presented thus:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in `Mono-spaced Roman` but are presented and highlighted as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }

}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

A note is a tip or shortcut or alternative approach to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring Important boxes won't cause data loss but may cause irritation and frustration.

Warning

A Warning should not be ignored. Ignoring warnings will most likely cause data loss.

Chapter 1. Introduction to RestComm

RestComm is a turnkey Cloud Communications platform based on Open Source building blocks from the team behind Mobicents. Restcomm offers a clean room implementation of the Twilio.com APIs and much more.

Restcomm can be integrated with VoIP and legacy SS7 network providers either in the cloud or via on-premise Resource Adaptors.

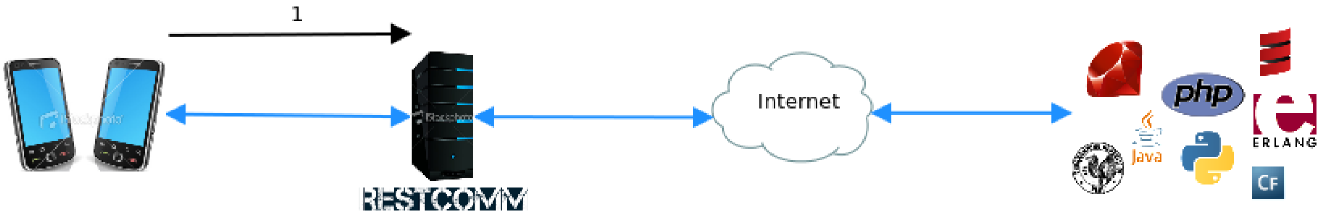
1.1. How it Works

In order to demonstrate how RestComm works we will go over an application that instructs RestComm to answer a phone call, say "Hello World" and finally hang up the call.

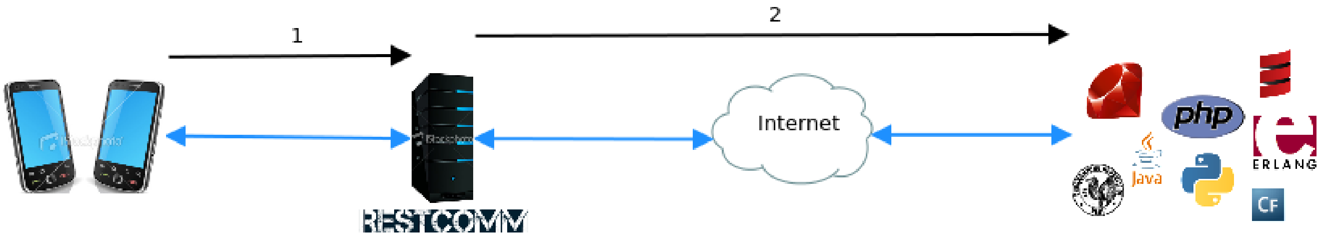
Note

A more thorough explanation of the RestComm Markup Language is available in Chapter 3. RestComm Markup Language

In this particular example, the first thing that must happen is that TelScale RestComm receives a call.

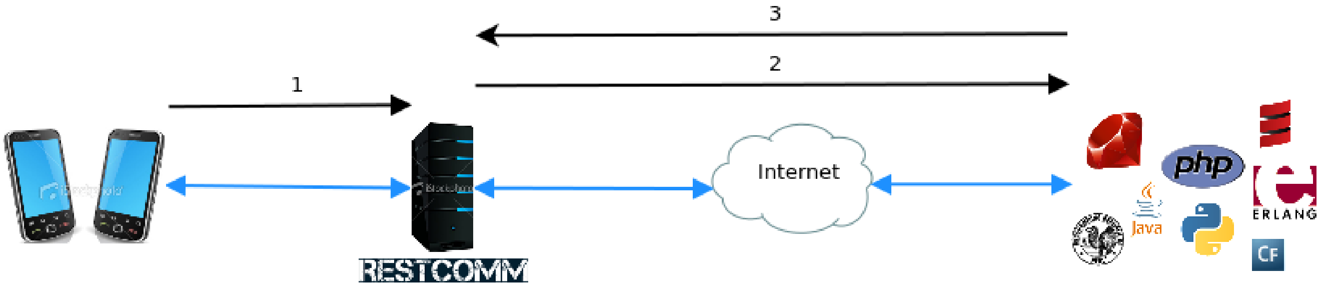


After TelScale RestComm can confirm that the call is destined for an application that it handles. TelScale RestComm calls out to your application for instructions on how to handle the new incoming call. Keep in mind that when TelScale RestComm calls out to your application it provides useful information such as from what number the call was dialed, to what number the call is destined, caller ID information, etc.

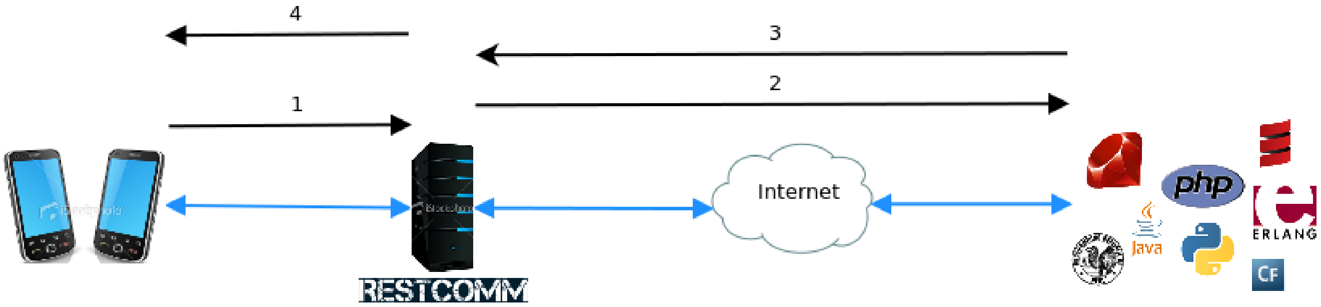


Once your application responds with instructions on how to handle the call, TelScale RestComm gets busy executing the provided instruction set.

```
<Response>
  <Say>Hello World!</Say>
</Response>
```



In this case TelScale RestComm synthesizes the text to speech, says "Hello World!" to the caller and hangs up.



Keep in mind that before you continue you should configure your ASR, Fax, SMS, and TTS plugins in order to have everything function as you would expect it to.

Chapter 2. Getting Started and General Configuration

2.1. Quick Start Guide

RestComm is a robust, powerful platform that will facilitate building comprehensive real-time communication solutions. The steps below will help you get started with ease.

Warning

On a production system, please take into consideration the following:

MMS can get into a bad state and report "Too many open files" error because of Linux default value on the total number of file descriptors.

You can go to the following sites to see how to increase the maximum number of open files in your server:

http://www.cyberciti.biz/faq/linux-increase-the-maximum-number-of-open-files/

http://amitbhayani.blogspot.fr/2010/01/javanetsocketexception-too-many-open.html

Running RestComm on JBoss

JBoss Security

Running a secure application is dependent on multiple factors. Restcomm runs on JBoss which implies that system security implemention can be handled at JBoss level. Please see the link below on how you can make your server environment more secure. https://community.jboss.org/wiki/SecureJboss?_sscc=t

- Download `TelScale-Restcomm-JBoss-AS7-XX.XX.GA.zip`
- Using a terminal of your choice, extract the content of `TelScale-Restcomm-XX.XX.GA.zip` to a local directory on your computer. The root directoy into which you extract the content of the .zip file will be referred to as $RESTCOMM_HOME.
- Go to `$RESTCOMM_HOME/telscale-media/telscale-media-server/bin`
- change the permission of the `run.sh` as follows

- sudo chmod +x ./run.sh
- then start the TelScale Media Server as follows
- sudo ./run.sh
- If all is correctly started you will see the following at the end of the bash terminal window

```
[MainDeployer] [[[[[[[[ Mobicents Media Server:
 release.version=3.0.0.FINAL Started ]]]]]]]]
```

- Open another terminal and proceed as follows:
- Go to `$RESTCOMM_HOME/bin`
- change the permissions of all the `.sh` files in the bin directory as follows
- `sudo chmod +x ./*.sh`
- Start RestComm by running the following command
- `sudo ./standalone.sh -c standalone-sip.xml`
- If RestComm is correctly started you will see the following at the end of the terminal

```
INFO  [Version] TelScale Sip Servlets 6.1.3.GA-TelScale (build: Git
 Hash=r8947f2732ee64c76566ed6c0b236204c048538e1 date=201306131639)
 Started.
17:30:05,854 INFO  [Version]
==============================================================================
==
  ==
==                      Thank you for running TelScale
  ==
==    Carrier Grade Communications Platform by the creators of Mobicents
  ==
==                    Copyright 2011-2013 Telestax, Inc.
  ==
==                         http://www.telestax.com/
  ==
==
  ==
==============================================================================
```

Running RestComm on Tomcat

- Download `TelScale-Restcomm-Tomcat-XX.XX.GA.zip`

- Using a terminal of your choice, extract the content of `TelScale-Restcomm-Tomcat-XX.XX.GA.zip` to a local directory on your computer. The root directoy into which you extract the content of the .zip file will be referred to as $RESTCOMM_HOME.

- Go to `$RESTCOMM_HOME/telscale-media/telscale-media-server/bin`

- change the permission of the `run.sh` as follows

- sudo chmod +x ./run.sh

- then start the TelScale Media Server as follows

- sudo ./run.sh

- If all is correctly started you will see the following at the end of the bash terminal window

```
[MainDeployer] [[[[[[[[ Mobicents Media Server:
 release.version=3.0.0.FINAL Started ]]]]]]]]
```

- Open another terminal and proceed as follows:

- Go to `$RESTCOMM_HOME/bin`

- change the permissions of all the `.sh` files in the bin directory as follows

- `sudo chmod +x ./*.sh`

- The content of the bin directory should be similar to the list below

```
bootstrap.jar                  digest.sh              tomcat-juli.jar
catalina.bat                   setclasspath.bat       tomcat-native.tar.gz
catalina.sh                    setclasspath.sh        tool-wrapper.bat
catalina-tasks.xml             shutdown.bat           tool-wrapper.sh
commons-daemon.jar             shutdown.sh            version.bat
commons-daemon-native.tar.gz   startup.bat            version.sh
cpappend.bat                   startup.sh
digest.bat                     telestax-license.xml
```

  Start RestComm by running the following command

- `sudo ./catalina.sh run`

- If RestComm is correctly started on Tomcat, you will see an output similar to the one below

```
Started.
```

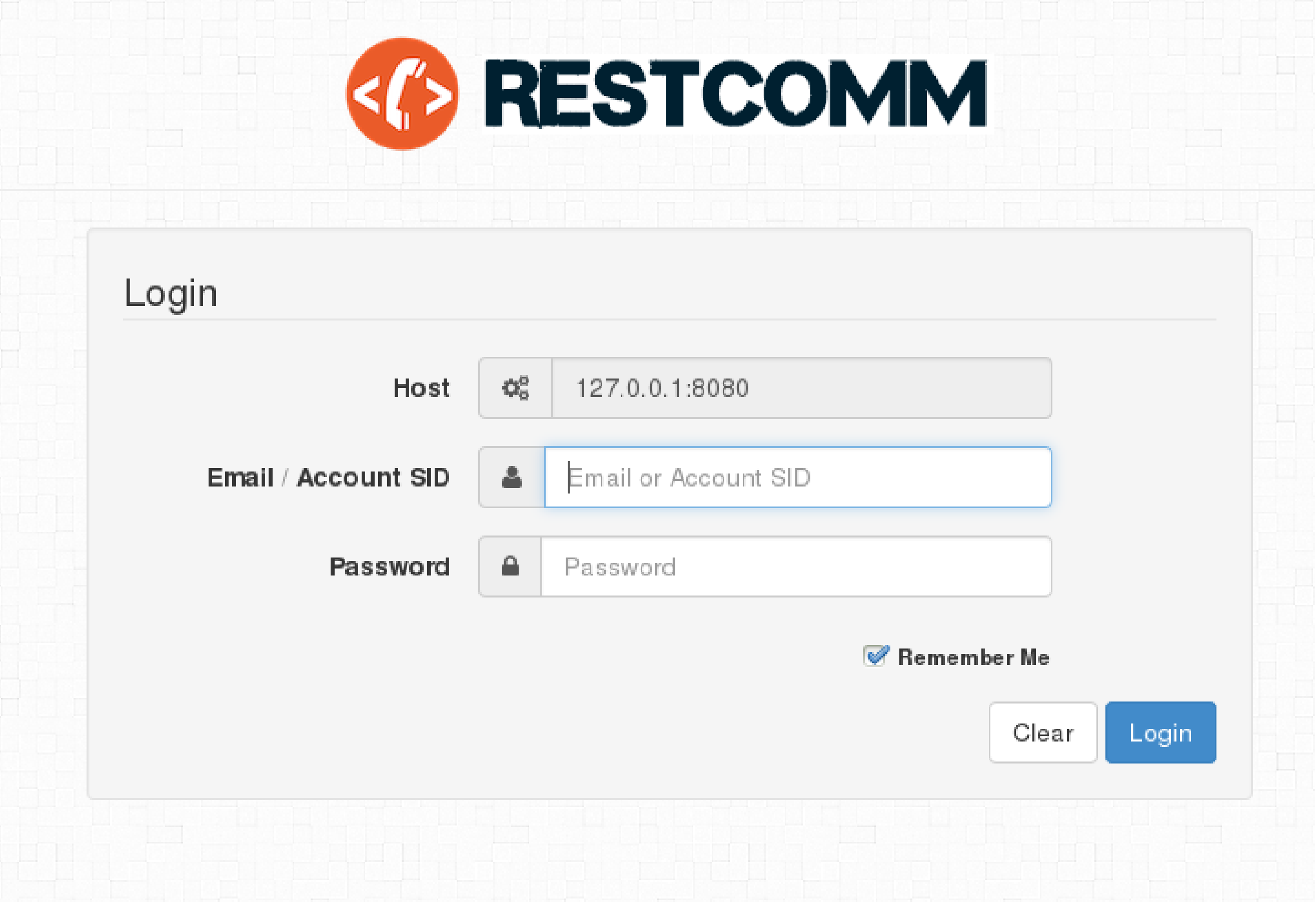
```
2013-08-15 19:02:46,903 INFO  [Version] (main)
==============================================================================
==
  ==
==                      Thank you for running TelScale
  ==
==    Carrier Grade Communications Platform by the creators of Mobicents
  ==
==                    Copyright 2011-2013 Telestax, Inc.
  ==
==                         http://www.telestax.com/
  ==
==
  ==
==============================================================================
Aug 15, 2013 7:02:50 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 14090 ms
```

2.2. Login into Restcomm and Changing the Default Password

Before you start using Restcomm, you must change the default password. Please follow the steps below

```
Open your browser and go to http://127.0.0.1:8080/restcomm-management

Enter the following information:
   - Email/Account SID = administrator@company.com
   - Password = RestComm
   - Press the  Login button
```

You will see a screen similar to the one below



Login

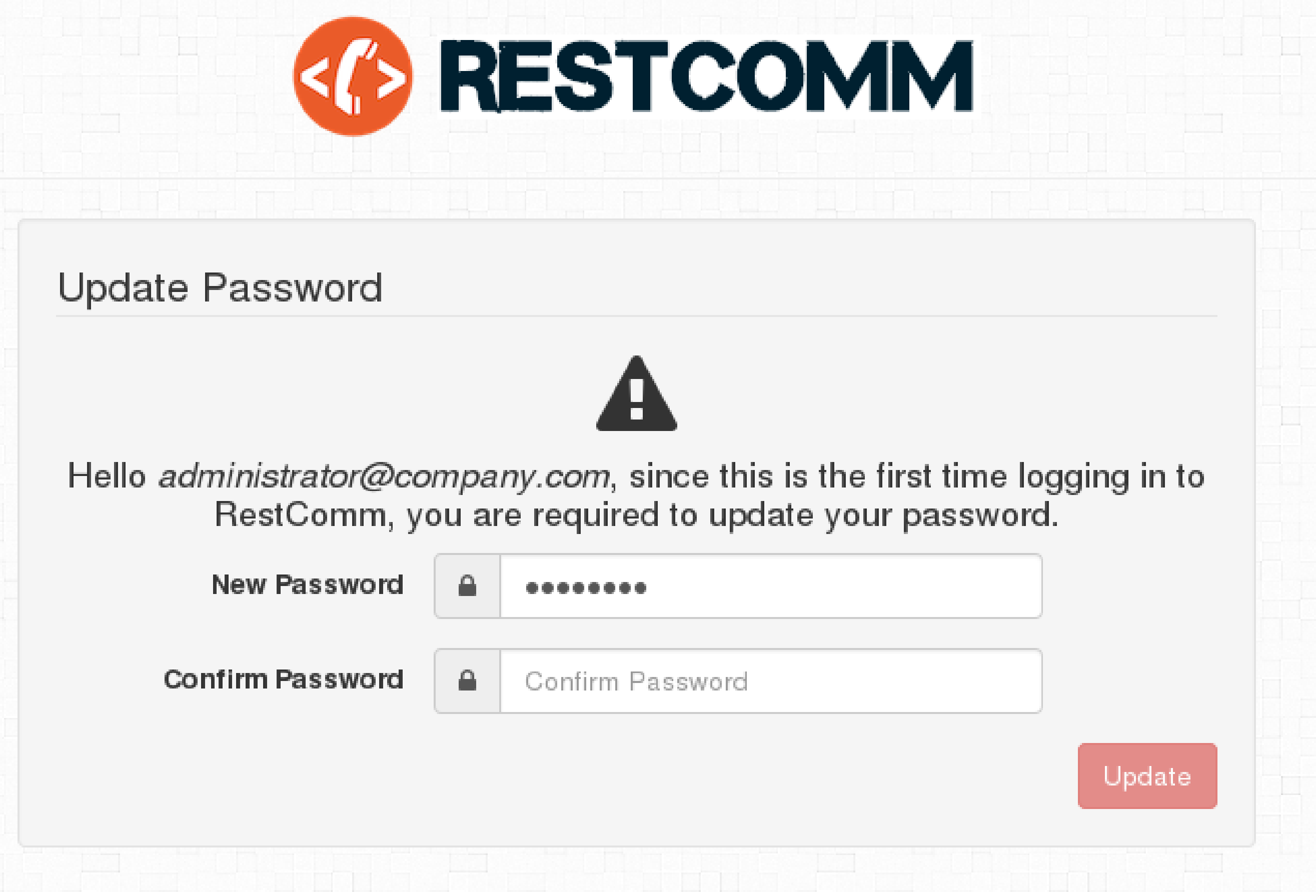
Host: 127.0.0.1:8080

Email / Account SID:

Password:

☑ Remember Me

In the screen below, change the password and update.



Update Password

Hello *administrator@company.com*, since this is the first time logging in to RestComm, you are required to update your password.

New Password: ●●●●●●●●●

Confirm Password: Confirm Password

Once the password has been changed, you can now log into the Admin Management interface and start using Restcomm.

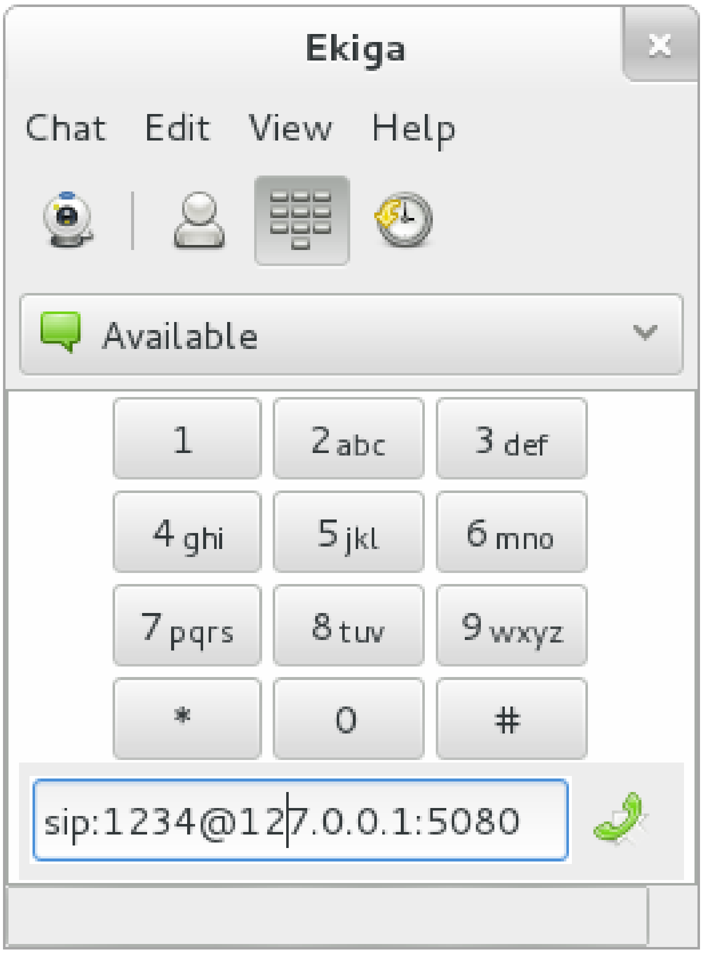
2.3. Testing the Demo Applications

Restcomm comes prepackaged with multiple example applications designed to help you quickly get started. For more demos and configuration details please visit www.telestax.com/doc [http://www.telestax.com/doc/]

2.3.1. Demo 1 - Testing the Play Verb

2.3.1. Demo 1 - Testing the Play Verb

Start a SIP phone (see below) and dial `1234@127.0.0.1:5080`. You will hear a welcome message.



sip:1234@127.0.0.1:5080

Warning

Some SIP phones have codec incompatibility issues, in the above example, we used Ekiga. You may also try Jitsi or Sflphone.

2.3.1.1. Quick Start Guide

RestComm is a robust, powerful platform that will facilitate building comprehensive real-time communication solutions. The steps below will help you get started with ease.

Warning

On a production system, please take into consideration the following:

MMS can get into a bad state and report "Too many open files" error because of Linux default value on the total number of file descriptors.

You can go to the following sites to see how to increase the maximum number of open files in your server:

http://www.cyberciti.biz/faq/linux-increase-the-maximum-number-of-open-files/

http://amitbhayani.blogspot.fr/2010/01/javanetsocketexception-too-many-open.html

Running RestComm on JBoss

JBoss Security

Running a secure application is dependent on multiple factors. Restcomm runs on JBoss which implies that system security implemention can be handled at JBoss level. Please see the link below on how you can make your server environment more secure. https://community.jboss.org/wiki/SecureJboss?_sscc=t

- Download `TelScale-Restcomm-JBoss-AS7-XX.XX.GA.zip`
- Using a terminal of your choice, extract the content of `TelScale-Restcomm-XX.XX.GA.zip` to a local directory on your computer. The root directoy into which you extract the content of the .zip file will be referred to as $RESTCOMM_HOME.
- Go to `$RESTCOMM_HOME/telscale-media/telscale-media-server/bin`
- change the permission of the `run.sh` as follows
- sudo chmod +x ./run.sh
- then start the TelScale Media Server as follows
- sudo ./run.sh
- If all is correctly started you will see the following at the end of the bash terminal window

```
     [MainDeployer] [[[[[[[[[ Mobicents Media Server:
release.version=3.0.0.FINAL
     Started ]]]]]]]]]
```

- Open another terminal and proceed as follows:
- Go to `$RESTCOMM_HOME/bin`
- change the permissions of all the `.sh` files in the bin directory as follows
- `sudo chmod +x ./*.sh`

- Start RestComm by running the following command
- `sudo ./standalone.sh -c standalone-sip.xml`
- If RestComm is correctly started you will see the following at the end of the terminal

```
    INFO [Version] TelScale Sip Servlets 6.1.3.GA-TelScale (build: Git
    Hash=r8947f2732ee64c76566ed6c0b236204c048538e1 date=201306131639)
    Started.
    17:30:05,854 INFO [Version]

==============================================================================
    == ==
    == Thank you for running TelScale ==
    == Carrier Grade Communications Platform by the creators of Mobicents
    ==
    == Copyright 2011-2013 Telestax, Inc. ==
    == http://www.telestax.com/ ==
    == ==

==============================================================================
```

Running RestComm on Tomcat

- Download `TelScale-Restcomm-Tomcat-XX.XX.GA.zip`
- Using a terminal of your choice, extract the content of `TelScale-Restcomm-Tomcat-XX.XX.GA.zip` to a local directory on your computer. The root directoy into which you extract the content of the .zip file will be referred to as $RESTCOMM_HOME.
- Go to `$RESTCOMM_HOME/telscale-media/telscale-media-server/bin`
- change the permission of the `run.sh` as follows
- sudo chmod +x ./run.sh
- then start the TelScale Media Server as follows
- sudo ./run.sh
- If all is correctly started you will see the following at the end of the bash terminal window

```
     [MainDeployer] [[[[[[[[[ Mobicents Media Server:
release.version=3.0.0.FINAL
     Started ]]]]]]]]]
```

- Open another terminal and proceed as follows:
- Go to `$RESTCOMM_HOME/bin`
- change the permissions of all the `.sh` files in the bin directory as follows
- `sudo chmod +x ./*.sh`
- The content of the bin directory should be similar to the list below

```
bootstrap.jar digest.sh tomcat-juli.jar
catalina.bat setclasspath.bat tomcat-native.tar.gz
catalina.sh setclasspath.sh tool-wrapper.bat
catalina-tasks.xml shutdown.bat tool-wrapper.sh
commons-daemon.jar shutdown.sh version.bat
commons-daemon-native.tar.gz startup.bat version.sh
cpappend.bat startup.sh
digest.bat telestax-license.xml
```

Start RestComm by running the following command

- `sudo ./catalina.sh run`
- If RestComm is correctly started on Tomcat, you will see an output similar to the one below

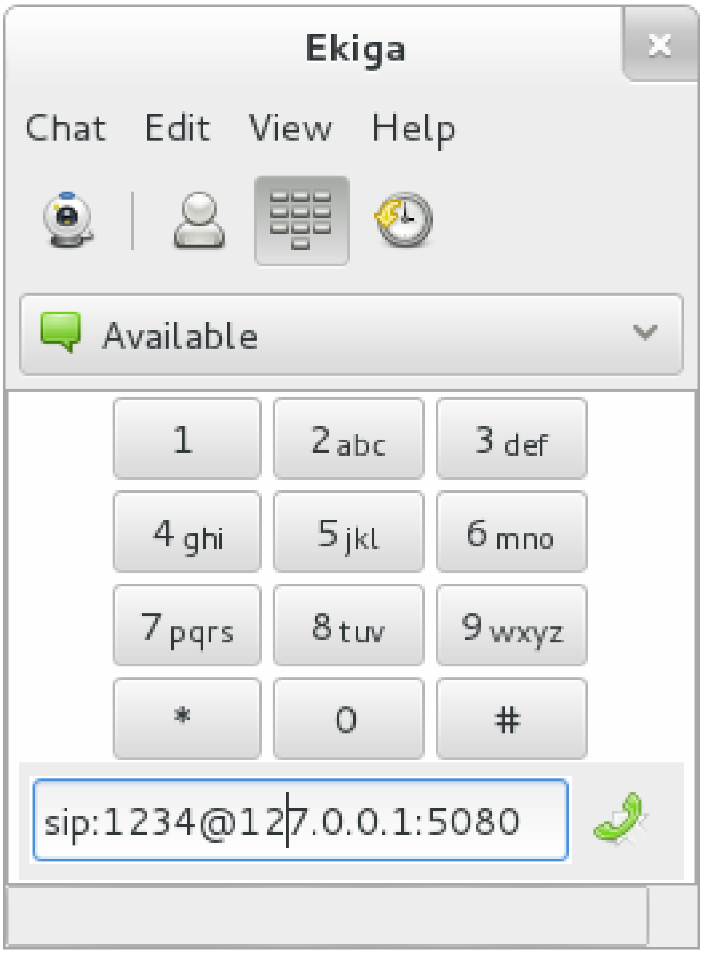
```
    Started.
    2013-08-15 19:02:46,903 INFO [Version] (main)

==============================================================================
    == ==
    == Thank you for running TelScale ==
    == Carrier Grade Communications Platform by the creators of Mobicents
    ==
    == Copyright 2011-2013 Telestax, Inc. ==
    == http://www.telestax.com/ ==
    == ==

==============================================================================
    Aug 15, 2013 7:02:50 PM org.apache.catalina.startup.Catalina start
    INFO: Server startup in 14090 ms
```

2.3.1.2. Testing the Demo Applications

Restcomm comes prepackaged with multiple example applications designed to help you quickly get started.

2.3.1.2.1. Demo 1 - Testing the Play Verb

Start a SIP phone (see below) and dial `1234@127.0.0.1:5080`. You will hear a welcome message.



sip:1234@127.0.0.1:5080

Warning

Some SIP phones have codec incompatibility issues, in the above example, we used Ekiga. You may also try Jitsi or Sflphone.

```
The application bound to the number 1234 can be found at
 <filename>$RESTCOMM_HOME/standalone/deployments/restcomm.war/demos/hello-play.xml</filename>.
```

2.3.1.2.2. Demo 2 - Testing Say Verb

You must first activate Text-to-Speech before you can proceed.

You must get an API key from http://www.voicerss.org [http://www.voicerss.org/] in order to proceed with this section. You can register for a free account and an API key will be emailed to you. Once you have the API key, open the `$RESTCOMM_HOME/standalone/deployments/restcomm.war/WEB-INF/conf/restcomm.xml` file and find the speech-synthesizer VoiceRSS section. Add your API key as shown below and restart RestComm

```
<speech-synthesizer
  class="org.mobicents.servlet.restcomm.tts.VoiceRSSSpeechSynthesizer">
  <service-root>http://api.voicerss.org</service-root>
  <apikey>2901c0aXXXXXXXXXXXXXX</apikey>
```

Start a SIP phone dial `1235@127.0.0.1:5080` . You will hear a welcome message in multiple languages.

```
The application bound to the number 1235 can be found at
 $RESTCOMM_HOME/standalone/deployments/restcomm.war/demos/hello-world.xml.
```

2.3.1.2.3. Demo 3 - Testing Gather Verb

This demo creates a simple IVR system

Activate DTMF using Ekiga

Make sure your DTMF setting in Ekiga is set to RFC2833. In order to set it correctly, go to the menu Edit->Preference->Protocols->SIP Settings->DTMF Mode

You may also use SFLPHONE instead of Ekiga

Start a SIP phone dial `1236@127.0.0.1:5080` . You will hear a message asking you to enter a digit and press star. If the digit is correctly received, Restcomm will replay the number you entered.

```
The application bound to the number 1236 can be found at
```

```
<filename>$RESTCOMM_HOME/standalone/deployments/restcomm.war/demos/gather/hello-gather.xml</
filename>.
and
<filename>$RESTCOMM_HOME/standalone/deployments/restcomm.war/demos/gather/gather.jsp</
filename>.
```

2.3.1.2.4. Demo 4 - Testing the Dial Sip Noun

This demo makes a call from one SIP phone to another. The Demo uses the SIP noun. You can calll any SIP account. All you have to do is change the content of the `dial-sip.xml`

In order to use this demo, you may use the default accounts, Alice and Bob, and register them on two separate SIP phones. Start both SIP phones and make sure Alice and Bob are registerd with the password (1234). These users come pre-configured with Restcomm for test purposes.

> **Start Two SIP Phone Sessions**
>
> If you are using the SIP sflphone here is what to do:
>
> Start first instance ex. - sflphone
>
> Start second instance ex. - sudo sflphone

In the application `dial-sip.xml` you can change the default to `sip:alice@127.0.0.1:5061?`

This will allow you to make a call to Alice. Note that Alice must be registered on port 5061 for the call to succeed.

From the the phone on which Bob is registerd, dial the number `1237` . The phone on which Alice is registered will ring and the connection will be made when you answer the call.

```
The application bound to the number 1237 can be found at
$RESTCOMM_HOME/standalone/deployments/restcomm.war/demos/dial/sip/dial-sip.xml.
```

2.3.1.2.5. Demo 5 - Testing the Client Noun

This demo makes a call from one SIP Client to Another. The demo uses the Client noun

In order to use this demo, you must have user Alice and Bob registered on two separate SIP phones. Start both SIP phones and make sure Alice and Bob are registerd with the password (1234). These users come pre-configured with Restcomm for test purposes.

Start Two SIP Phone Sessions

You can start the second instance of your SIP phones by prepending the executable with `sudo` . If you are using the SIP sflphone here is what to do:

Start first instance ex. - sflphone

Start second instance ex. - sudo sflphone

From the phone on which Bob is registerd, dial the number `1238` . The phone on which Alice is registered will ring and the connection will be made when you answer the call.

```
The application bound to the number 1238 can be found at
<filename>$RESTCOMM_HOME/standalone/deployments/restcomm.war/demos/dial/client/dial-
client.xml</filename>.
```

Advanced Examples

Please go to this chapter for more examples Chapter 7, *Advanced RestComm Examples* .

2.3.1.2.6. Demo 6 - Testing Conference Noun

This demo Lets a user join a conference as a moderator and the other user as a participant. The participant will dial `1310` and will hear a hold music. The moderator will dial `1311` and the hold music will stop and the conference will be started.

Most SIP phones will require you to register before you can make a call. You can use the default accounts, Alice and Bob with password (1234)to register.

Start Two SIP Phone Sessions

You can start the second instance of your SIP phones by prepending the executable with `sudo` . If you are using the SIP sflphone here is what to do:

Start first instance ex. - sflphone

Start second instance ex. - sudo sflphone

From the phone on which Bob is registerd, dial the number `1310` . From the phone on which Alice is registered, dial `1311`

```
The application is bound to the number 1310 and 1311 can be found at

http://127.0.0.1:8080/restcomm/demos/dial/conference/dial-conference.xml

and at

http://127.0.0.1:8080/restcomm/demos/dial/conference/dial-conference-moderator.xml
```

Advanced Examples

Please go to this chapter for more examples Chapter 7, *Advanced RestComm Examples* .

2.3.1.2.7. Demo 7 - RVD Say Verb

This example is built using the Restcomm Visual Designer. You can register a SIP phone with the user bob or alice and password (1234). Then dial `1239`. You will hear a welcome message.

```
The application is bound to the number 1239. The link to the rvdSayVerbDemo is below:

http://127.0.0.1:8080/restcomm-rvd/#/designer/rvdSayVerbDemo

All project created with Restcomm Visual Designer are stored in the directory:

$RESTCOMM_HOME/standalone/deployments/restcomm-rvd.war/workspace
```

2.3.1.2.8. Demo 8 - RVD Gather Verb

This demo uses the Gather verb to get user input and map it to the corresponding modules. You can register a SIP phone with the user bob or alice and password (1234). Then, dial `1240`. You will hear a welcome message asking you to press 1 for Sales or 2 for Technical Support.

```
The application is bound to the number 1240. The link to the rvdSayVerbDemo is below:
```

```
http://127.0.0.1:8080/restcomm-rvd/#/designer/rvdCollectVerbDemo

All project created with Restcomm Visual Designer are stored in the directory:

$RESTCOMM_HOME/standalone/deployments/restcomm-rvd.war/workspace
```

2.3.1.2.9. Demo 9 - RVD External Services

This demo uses the External Services to Restcomm to an external program for more complex solutions. There are two parts to this demo application, the RVD app and the external servlet `http://127.0.0.1:8080/RvdExternalServicesDemo/rvdESdemo` which manages the logic. You can register a SIP phone with the user bob or alice and password (1234). Then, dial `1241`. You will be asked to enter a customer ID. You can enter number `11 or 12`. You will hear a corresponding welcome message depending on the customer ID you entered.

```
 The application is bound to the number 1241. The link to the rvdSayVerbDemo is below:

 http://127.0.0.1:8080/restcomm-rvd/#/designer/rvdESDemo

 All project created with Restcomm Visual Designer are stored in the directory:

 $RESTCOMM_HOME/standalone/deployments/restcomm-rvd.war/workspace

 The RvdExternalServicesDemo.war file is deployed in the directory $RESTCOMM_HOME/standalone/
deployments/
```

2.3.2. Detailed Configuration

In order for TelScale RestComm to function properly it must first be configured. In this chapter we will cover the settings for TelScale RestComm and it's respective plug-ins. In order to make TelScale RestComm easy to manage all of the configuration is done in only one file. The file is located at `RESTCOMM_HOME/WEB-INF/webapps/restcomm/conf/restcomm.xml` and is composed of the sections that follow.

> **Note**
>
> TelScale RestComm provides a default set of plug-ins for the storage engine, fax service, SMS aggregator, automatic speech recognition, and speech synthesis. All these services are pluggable and TelScale RestComm is not limited to any service provider or software platformz.

2.3.3. Runtime Settings

The runtime-settings are used by TelScale RestComm at runtime to customize it's behavior. A list of the runtime settings and a description is provided below.

Table 2.1. Runtime Settings

Element	Description
api-version	The version of the TelScale RestComm API that we will be executing by default.
prompts-uri	The location where the audio prompts are located.
cache-path	The local path the cache folder.
cache-uri	The HTTP URI to the cache folder.
recordings-path	The local path to the folder containing the recordings.
recordings-uri	The HTTP URI to the folder containing the recordings.
error-dictionary-uri	The HTTP URI to the TelScale RestComm error dictionary.
external-ip	The IP to use for out-bound SIP REGISTER requests. This is useful when you want to report a different IP than the one TelScale RestComm picked by default.
use-to	If set to true TelScale RestComm will use the To header to determine the destination. If set to false TelScale RestComm will use the Request URI to determine the destination.
outbound-proxy-user	The user name used to authenticate with the outbound proxy. (Optional)
outbound-proxy-password	The password used to authenticated with the outbound proxy. (Optional)
outbound-proxy-uri	The SIP URI to the outbound proxy. Note: Do not prepend 'sip:' to the proxy uri. If necessary the port can be included, for example alice@localhost:5080
fallback-outbound-proxy-user	The user name used to authenticate with the fallback outbound proxy. (Optional)
fallback-outbound-proxy-password	The password used to authenticated with the fallback outbound proxy. (Optional)

Element	Description
fallback-outbound-proxy-uri	The SIP URI to the fallback outbound proxy. Note: Do not prepend 'sip:' to the proxy uri. If necessary the port can be included, for example alice@localhost:5080
allow-fallback	Set this to true so Restcomm will fallback to the backup proxy in case of failed calls
max-failed-calls	Maximum number of failed call before switching from primary to fallback outbound proxy
allow-fallback-to-primary	Allow fallback from Backup to Primary proxy in case fail called at backup proxy.

TelScale Restcomm Resource Security. The security model is based on role based access control. TelScale RestComm defines a set of permissions that can be defined for each role. There are two predefined roles Administrator and Developer. The Developer role can be configured or removed all together depending on your needs but the Administrator role can not be changed. The Administrator role can not be modified or removed and it has access to every resource.

First we will define the list of permissions and what they mean.

Table 2.2. TelScale RestComm Permissions

Permission	Description
Create	The role has access to create a type resource.
Read	The role has access to read a type of resource.
Modify	The role has access to modify a type of resource.
Delete	The role has access to delete a type of resource.

These permissions apply to every resource exposed by the Restful APIs. Once a role is defined it can be specified for new Account resources that are created.

Note

If no role is specified when creating a new Account resource then the new Account will inherit the security role of the account that created it. If this role is the Administrator role the system may become compromised.

Wildcard Permission. The asterisk '*' is a wildcard that means grant all permissions and can be used in place of typing out all the permissions.

To see how this all comes together please check out the `restcomm.xml` configuration file.

2.3.4. VoIP Innovations Restful API Access

The default auto provisioning API used by TelScale RestComm.

Table 2.3. VoIP Innovations Settings

Element	Description
login	The user name configured in the VoIP Innovations Portal under API users.
password	The password configured in the VoIP Innovations Portal under API users.
endpoint	The end point ID of the end point that will be used by TelScale RestComm
uri	The path to the VoIP Innovations service end point.

2.3.5. Dao Manager

The data access object manager is used to gain access to the data store that TelScale RestComm will use.

The default data access object manager is based on MyBatis and provides access to RDBM systems.

Table 2.4. MyBatis Settings

Element	Description
configuration-file	The path to the mybatis.xml configuration file.
data-files	The path to the data files used to store the database tables.
sql-files	The path to the XML files containing the SQL statements that will be executed by the database.

2.3.6. Media Server Manager

The media server manager is responsible for all the media servers managed by TelScale RestComm.

Table 2.5. Media Server Manager Settings

Property	Description
local-address	The local IP address for the MGCP stack.
local-port	The local port for the MGCP stack.
remote-address	The IP address for the media server.
remote-port	The port for the media server.
response-timeout	In milliseconds the maximum amount of time to wait for a response from the media server before abandoning the request. This does NOT apply to RQNT/NOTIFY request/ response.
external-address	Sometimes there is interest to use a different address in the SDP than the IP address the media server is reporting. This parameter if set tells TelScale RestComm to patch the Connection attribute in the SDP on behalf of the media server to the specified IP address. Note: TelScale RestComm will only do NAT resolution when necessary so if your server already has a routable IP address setting this parameter will have no effect.

2.3.7. SMS Aggregator

The SMS aggregator is responsible for the handling of SMS messages on behalf of TelScale RestComm.

Requirements. The default SMS aggregator is SMS over IP as used by VoIP Innovations. Therefore, an account must be created @ http://www.voipinnovations.com [http://www.voipinnovations.com/] before any text messages can be sent.

Below is a list of settings that must be configured to send text messages.

Table 2.6. SMS Aggregator Settings

Element	Description
outbound-prepend-string	A string that you would like to prepend to the destination address. For example, to use service from VoIP Innovations a '#' must be prepended to every destination address.
outbound-endpoint	The SIP endpoint to which outbound SMS messages should be sent. Note: Do not prepend sip: to the endpoint.

2.3.8. Fax Service

The fax service sends faxes out on behalf of TelScale RestComm.

Requirements. The default fax service is provided by Interfax. Therefore, an account must be created @ http://www.interfax.net [http://www.interfax.net/] before any faxes can be sent.

Table 2.7. Fax Service Settings

Element	Description
user	The user name used to authenticate with Interfax.
password	The password used to authenticate with Interfax.

2.3.9. Speech Recognizer

This speech recognizer turns speech into text on behalf of TelScale Restcomm.

Requirements. The default speech recognizer uses the ASR service provided by iSpeech. Therefore, an account must be created @ http://www.ispeech.org [http://www.ispeech.org/] before any speech can be converted to text.

Table 2.8. Speech Recognizer Settings

Element	Description
api-key	The Web API key provided by iSpeech. This is used as an authentication token for the service.

2.3.10. Speech Synthesizer

The speech synthesizer turns text to speech on behalf of TelScale RestComm.

Requirements. The default speech synthesizer uses the TTS service provided by VoiceRSS. Therefore, an account must be created @ http://www.voicerss.org [http://http://www.voicerss.org/] before any text can be converted to speech.

Table 2.9. VoiceRSS Speech Synthesizer Settings

Element	Description
service-root	The VoiceRSS service root URI.
apikey	The API key used to authenticate with the VoiceRSS.

Element	Description
languages	A list of male and female speakers for different languages. These can be replaced with other alternatives provided by VoiceRSS.

Chapter 3. RestComm Markup Language

The RestComm Markup Language (RCML) is composed of a set of XML tags that can be used to instruct RestComm on how to handle an on-going phone call. RestComm applications are built by combining these XML verbs and nouns in a way that is sensible for a given set of application requirements. In this chapter we will discuss what a request from RestComm to your application looks like as well as what the response from your application should look like. Then, we will dive in to how each verb and noun in the XML instruction set is used.

3.1. RestComm Request

How RestComm Passes Data to Your Application. The way RestComm passes data to you application depends on the request method for the given URI. If the request method is GET then the data is passed in the query string or the part after the question mark. If the request method is POST then the data is sent a multi-part form data just like when a browser submits a form.

RestComm Voice Request. Any time RestComm makes a request to you applications it will include the following data as request parameters.

Table 3.1. Request Parameters

Parameter	Description
CallSid	The unique identifier for this call.
AccountSid	Your account id.
From	The phone number of the originator of the call.
To	The phone number of the call recipient.
CallStatus	The status of the call. The possible values are queued, ringing, in-progress, completed, busy, failed, and no-answer.
ApiVersion	The version of the RestComm API used to handle this call.
Direction	The direction of the call. The possible values are inbound and outbound-dial.
CallerName	The caller ID for the caller in the case of inbound calls.

3.2. Your Response

In your response to the request from RestComm you want to provide RCML that will instruct RestComm on how to handle the current call.

MIME Types. RestComm supports the MIME types described in the table below.

Table 3.2. Supported MIME Types

Parameter	Description
text/xml, application/xml	RestComm interprets the returned document as an XML instruction set.

Note

When your application returns the RCML document the root element of the document must always be <Response> or the parser will complain.

3.3. Say

The <Say> verb is used to synthesize text to speech and play it to the remote party. The voices supported depends on the TTS Service provider plug-in. Below are the voices for our default TTS service provider plug-in which uses Acapela Voice and VoiceRSS (default) as a Service.

Table 3.3. Say Attributes

Name	Allowed Values	Default Value
voice	man, woman	man
language	bf, bp, en, en-gb,cf, cs, dan, fi es, fr, de, el, it, nl, no, pl, pt, ru, ar, ca, sv, tr	en
loop	integer > 1	1

voice. The 'voice' attribute allows you to select the gender of the voice used to synthesize the text to speech for playback.

language. The 'language' attribute allows you pick a specific language for speech synthesis. RestComm currently supports languages 'bf' (Belgium-French), 'bp' (Brazilian-Portugues), 'en' (English), 'en-gb' (British-English), 'cf' (Canadian-French), 'cs' (Czech), 'dan' (Dannish), 'fi' (Finnish), 'es' (Spanish), 'fr' (French), 'de' (German), 'el' (Greek), 'it' (Italian), 'nl' (Netherlands-Dutch), 'no' (Norwegian), 'pl' (Polish), 'pt' (Portuguese), 'ru' (Russian), 'ar' (Saudi-Arabia Arabic), 'ca' (Spain Catalan), 'sv' (Swedish), and 'tr' (Turkish).

loop. The 'loop' attribute specifies how many times you'd like the text repeated. Specifying '0' will cause the the <Say> verb to loop until the call is hung up.

Nesting. The <Say> verb can not have any other verbs or nouns nested. Only text.

Examples. For an example of how to use the <Say> verb see below.

```
  <Response>
   <Say>Hello World</Say>
  </Response>
```

The example below shows how you can set the language, voice and loop parameters of the Say Verb.

```
  <Response>
 <Say voice="woman" language="fr" loop="3">Bienvenue à RestComm un projet parrainé par TeleStax</
Say>
  </Response>
```

Usefull Information

- When translating text to speech, the Say verb will make assumptions about how to pronounce numbers, dates, times, amounts of money and other abbreviations.
- When saying numbers, '12345' will be spoken as "twelve thousand three hundred forty-five." Whereas '1 2 3 4 5' will be spoken as "one two three four five."

3.4. Play

The <Play> verb is used to play an audio file to the remote party.

Table 3.4. Play Attributes

Name	Allowed Values	Default Value
loop	integer > 1	1

loop. The 'loop' attribute specifies how many times you'd like the audio file to be repeated. Specifying '0' will cause the the <Play> verb to loop until the call is hung up.

Table 3.5. Supported Audio Formats

MIME type	Description
audio/wav	wav format audio
audio/wave	wav format audio
audio/x-wav	wav format audio

Media Server Audio File Format

The recommended audio file format is linear(8000,16 bit , Mono).

Nesting. The <Play> verb can not have any other verbs or nouns nested. Only a URL.

Basic example. For an example of how to use the <Play> verb see below.

```
<Response>
 <Play>http://foobar.com/demo.wav</Play>
</Response>
```

3.5. Gather

The <Gather> verb "gathers" digits that a caller enters into his or her telephone keypad. When the caller is done entering digits, RestComm submits that digits to the provided 'action' URL in an HTTP GET or POST request. If no input is received before timeout, <Gather> falls through to the next verb in the RestComm document. You may optionally nest <Say>, <Play>, and <Pause> verbs within a <Gather> verb while waiting for input. This allows you to read menu options to the caller while letting her enter a menu selection at any time. After the first digit is received the audio will stop playing.

Table 3.6. Gather Attributes

Name	Allowed Values	Default Value
action	relative or absolute URL	current document URL
method	GET, POST	POST
timeout	positive integer	5 seconds
finishOnKey	any digit, #, *	#
numDigits	integer >= 1	unlimited

action. The 'action' attribute takes an absolute or relative URL as a value. When the caller has finished entering digits RestComm will make a GET or POST request to this URL including the parameters below. If no 'action' is provided, RestComm will by default make a POST request to the current document's URL.

Table 3.7. Request Parameters

Parameter	Description
Digits	The digits the caller pressed, excluding the finishOnKey digit.

method. The 'method' attribute takes the value 'GET' or 'POST'. This tells RestComm whether to request the 'action' URL via HTTP GET or POST.

timeout. The 'timeout' attribute sets the limit in seconds that RestComm will wait for the caller to press another digit before moving on and making a request to the 'action' URL. For example, if 'timeout' is '10', RestComm will wait ten seconds for the caller to press another key before submitting the previously entered digits to the 'action' URL. RestComm waits until completing the execution of all nested verbs before beginning the timeout period.

finishOnKey. The 'finishOnKey' attribute lets you choose one value that submits the received data when entered. For example, if you set 'finishOnKey' to '#' and the user enters '1234#', RestComm will immediately stop waiting for more input when the '#' is received and will submit "Digits=1234" to the 'action' URL. Note that the 'finishOnKey' value is not sent. The allowed values are the digits 0-9, '#' , '*' and the empty string (set 'finishOnKey' to ''). If the empty string is used, <Gather> captures all input and no key will end the <Gather> when pressed. In this case RestComm will submit the entered digits to the 'action' URL only after the timeout has been reached. The value can only be a single character.

numDigits. The 'numDigits' attribute lets you set the number of digits you are expecting, and submits the data to the 'action' URL once the caller enters that number of digits.

Nesting. You can nest the following verbs within <Gather>: <Say>, <Play>, <Pause>

Examples. For an example of how to use the <Gather> verb see below.

```
<Response>
 <Gather action="handle-user-input.php" numDigits="1">
  <Say>Welcome to TPS.</Say>
  <Say>For store hours, press 1.</Say>
  <Say>To speak to an agent, press 2.</Say>
  <Say>To check your package status, press 3.</Say>
 </Gather>
 <!-- If customer doesn't input anything, prompt and try again. -->
 <Say>Sorry, I didn't get your response.</Say>
 <Redirect>handle-incoming-call.xml</Redirect>
</Response>
```

3.6. Record

The <Record> verb records the caller's voice and returns to you the URL of a file containing the audio recording.

Table 3.8. Record Attributes

Name	Allowed Values	Default Value
action	relative or absolute URL	current document URL
method	GET, POST	POST
timeout	positive integer	5
finishOnKey	any digit, #, *	1234567890*#
maxLength	integer greater than 1 with the number of seconds to wait	3600 (1 hour)
transcribe	true, false	false
transcribeCallback	relative or absolute URL	none
playBeep	true, false	true

action. The 'action' attribute takes an absolute or relative URL as a value. When recording is finished RestComm will make a GET or POST request to this URL including the parameters below. If no 'action' is provided, <Record> will default to requesting the current document's URL. After making this request, RestComm will continue the current call using the RCML received in your response. Any RCML verbs occuring after a <Record> are unreachable. There is one exception: if RestComm receives an empty recording, it will not make a request to the 'action' URL. The current call flow will continue with the next verb in the current RCML document.

Table 3.9. Request Parameters

Parameter	Description
RecordingUrl	The URL of the recorded audio.
RecordingDuration	The time duration of the recorded audio.
Digits	The digits the caller pressed, excluding the finishOnKey digit.

method. The 'method' attribute takes the value 'GET' or 'POST'. This tells RestComm whether to request the URL via HTTP GET or POST.

timeout. The 'timeout' attribute tells RestComm to end the recording after a number of seconds of silence has passed.

finishOnKey. The 'finishOnKey' attribute lets you choose a set of digits that end the recording when entered. For example, if you set 'finishOnKey' to '#' and the caller presses '#', RestComm will immediately stop recording and submit 'RecordingUrl', 'RecordingDuration', and the '#' as parameters in a request to the 'action' URL. The allowed values are the digits 0-9, '#' and '*'. Unlike <Gather>, you may specify more than one character as a 'finishOnKey' value.

maxLength. The 'maxLength' attribute lets you set the maximum length for the recording in seconds.

transcribe. The 'transcribe' attribute tells RestComm that you would like a text representation of the audio of the recording.

transcribeCallback. The 'transcribeCallback' attribute is used in conjunction with the 'transcribe' attribute. It allows you to specify a URL to which RestComm will make an asynchronous POST request when the transcription is complete. This is not a request for RCML and the response will not change call flow, but the request will contain the standard RCML request parameters as well as 'TranscriptionStatus', 'TranscriptionText', 'TranscriptionUrl' and 'RecordingUrl'. If 'transcribeCallback' is specified, then there is no need to specify 'transcribe=true'. It is implied. If you specify 'transcribe=true' without a 'transcribeCallback', the completed transcription will be stored for you to retrieve later (see the REST API Transcriptions section), but RestComm will not asynchronously notify your application.

Table 3.10. Request Parameters

Parameter	Description
TranscriptionText	Contains the text of the transcription.
TranscriptionStatus	The status of the transcription attempt: either 'completed' or 'failed'.
TranscriptionUrl	The URL for the transcription's REST API resource.
RecordingUrl	The URL for the transcription's source recording resource.

playBeep. The 'playBeep' attribute allows you to toggle between playing a sound before the start of a recording.

Nesting. The <Record> verb can not have any other verbs or nouns nested.

Examples. For an example of how to use the <Record> verb see below.

```
<Response>
 <Record maxLength="30"/>
</Response>
```

3.7. Fax

The <Fax> verb sends a fax to some a fax machine.

Table 3.11. Fax Attributes

Name	Allowed Values	Default Value
to	phone number	see below
from	phone number	see below
action	relative or absolute URL	none
method	GET, POST	POST
statusCallback	relative or absolute URL	none

to. The 'to' attribute takes a valid E.164 phone number as a value. RestComm will send a fax to this number. When sending a fax during an incoming call, 'to' defaults to the caller. When sending an fax during an outgoing call, 'to' defaults to the called party. The value of 'to' must be a valid phone number.

from. The 'from' attribute takes a valid E.164 phone number as an argument. When sending a fax during an incoming call, 'from' defaults to the calling party. When sending a fax during an outgoing call, 'from' defaults to the called party. The value of 'from' must be a valid phone number.

action. The 'action' attribute takes a URL as an argument. After processing the <Fax> verb, RestComm will make a GET or POST request to this URL with the form parameters 'FaxStatus' and 'FaxSid'. Using an 'action' URL, your application can receive synchronous notification that the message was successfully enqueued. If you provide an 'action' URL, RestComm will use the RCML received in your response to the 'action' URL request to continue the current call. Any RCML verbs occuring after a <Fax> which specifies an 'action' attribute are unreachable. If no 'action' is provided, <Fax> will finish and RestComm will move on to the next RCML verb in the document. If there is no next verb, RestComm will end the phone call.

Table 3.12. Request Parameters

Parameter	Description
FaxSid	The Sid for the Sms message.
FaxStatus	The current status of the Sms message. Either 'sent' or 'failed'.

method. The 'method' attribute takes the value 'GET' or 'POST'. This tells RestComm whether to request the 'action' URL via HTTP GET or POST. This attribute is modeled after the HTML form 'method' attribute.

statusCallback. The 'statusCallback' attribute takes a URL as an argument. When the fax is actually sent, or if sending fails, RestComm will make an asynchronous POST request to this URL with the parameters 'FaxStatus' and 'FaxSid'. Note, 'statusCallback' always uses HTTP POST to request the given url.

Table 3.13. Request Parameters

Parameter	Description
FaxSid	The Sid for the fax message.
FaxStatus	The current status of the fax. Either 'sent' or 'failed'.

Nesting. The <Fax> verb can not have any other verbs or nouns nested.

Examples. For an example of how to use the <Fax> verb see below.

```
<Response>
 <Fax>This is a test fax.</Fax>
</Response>
```

3.8. Sms

The <Sms> verb sends an SMS message to a phone number during a phone call.

Table 3.14. Sms Attributes

Name	Allowed Values	Default Value
to	phone number	see below
from	phone number	see below
action	relative or absolute URL	none
method	GET, POST	POST
statusCallback	relative or absolute URL	none

to. The 'to' attribute takes a valid E.164 phone number as a value. RestComm will send an SMS message to this number. When sending an SMS during an incoming call, 'to' defaults to the caller. When sending an SMS during an outgoing call, 'to' defaults to the called party. The value of 'to' must be a valid phone number.

from. The 'from' attribute takes a valid E.164 phone number as an argument. When sending an SMS during an incoming call, 'from' defaults to the calling party. When sending an SMS during an outgoing call, 'from' defaults to the called party.

action. The 'action' attribute takes a URL as an argument. After processing the <Sms> verb, RestComm will make a GET or POST request to this URL with the form parameters 'SmsStatus' and 'SmsSid'. Using an 'action' URL, your application can receive synchronous notification that the message was successfully enqueued. If you provide an 'action' URL, RestComm will use the RCML received in your response to the 'action' URL request to continue the current call. Any

RCML verbs occuring after an <Sms> which specifies an 'action' attribute are unreachable. If no 'action' is provided, <Sms> will finish and RestComm will move on to the next RCML verb in the document. If there is no next verb, RestComm will end the phone call.

method. The 'method' attribute takes the value 'GET' or 'POST'. This tells RestComm whether to request the 'action' URL via HTTP GET or POST. This attribute is modeled after the HTML form 'method' attribute.

statusCallback. The 'statusCallback' attribute takes a URL as an argument. When the SMS message is actually sent, or if sending fails, RestComm will make an asynchronous POST request to this URL with the parameters 'SmsStatus' and 'SmsSid'. Note, 'statusCallback' always uses HTTP POST to request the given url.

Table 3.15. Request Parameters

Parameter	Description
SmsSid	The Sid for the Sms message.
SmsStatus	The current status of the Sms message. Either 'sent' or 'failed'.

Nesting. The <Sms> verb can not have any other verbs or nouns nested.

Examples. For an example of how to use the <Sms> verb see below.

```
<Response>
 <Sms>Hello World!</Sms>
</Response>
```

Sms Custom headers

Restcomm will check for any custom headers and make use of them.

In case of an incoming SMS (incoming SIP MESSAGE) Restcomm will scan the message for any headers starting with X- and if any found will be send to the application server with the request (part of the query string in the case of HTTP GET or part of the message body in the case of HTTP POST)

The application server can also provide some custom http headers (again starting with X-) along with the RCML response, that Restcomm will store and use them when creating the outgoing SMS message (in the case of RCML response is Sms).

3.9. Dial

The <Dial> verb connects the current caller to another phone. If the called party picks up, the two parties are connected and can communicate until one hangs up. If the called party does not pick up, if a busy signal is received, or if the number doesn't exist, the dial verb will finish.

Table 3.16. Dial Attributes

Name	Allowed Values	Default Value
action	relative or absolute URL	no default for <Dial>
method	GET, POST	POST
timeout	positive integer in seconds	30 seconds
timeLimit	positive integer (seconds)	14400 seconds (4 hours)
callerId	a valid phone number, or client identifier if you are dialing a <Client>.	Caller's callerId
record	true, false	false

action. The 'action' attribute takes a URL as an argument. When the dialed call ends, RestComm will make a GET or POST request to this URL including the parameters below. If you provide an 'action' URL, RestComm will continue the current call after the dialed party has hung up, using the RCML received in your response to the 'action' URL request. Any RCML verbs occuring after a <Dial> which specifies an 'action' attribute are unreachable. If no 'action' is provided, <Dial> will finish and RestComm will move on to the next RCML verb in the document. If there is no next verb, RestComm will end the phone call.

Table 3.17. Request Parameters

Parameter	Description
DialCallStatus	The outcome of the <Dial> attempt. See the DialCallStatus section below for details.
DialCallSid	The call sid of the new call leg. This parameter is not sent after dialing a conference.
DialCallDuration	The duration in seconds of the dialed call. This parameter is not sent after dialing a conference.
RecordingUrl	The URL of the recorded audio. This parameter is only sent if record="true" is set on the `Dial` verb, and does not include recordings from the `Record` verb or Record=True on REST API calls..

Table 3.18. DialCallStatus Values

Parameter	Description
completed	The called party answered the call and was connected to the caller.
busy	RestComm received a busy signal when trying to connect to the called party.
no-answer	The called party did not pick up before the timeout period passed.
failed	RestComm was unable to route to the given phone number. This is frequently caused by dialing a properly formated but non-existent phone number.
canceled	The call was canceled via the REST API before it was answered.

method. The 'method' attribute takes the value 'GET' or 'POST'. This tells RestComm whether to request the 'action' URL via HTTP GET or POST. This attribute is modeled after the HTML form 'method' attribute.

timeout. The 'timeout' attribute sets the limit in seconds that <Dial> waits for the called party to answer the call.

timelimit. The 'timeLimit' attribute sets the maximum duration of the <Dial> in seconds.

callerId. The 'callerId' attribute lets you specify the caller ID that will appear to the called party when RestComm calls. By default, when you put a <Dial> in your RCML response to RestComm's inbound call request, the caller ID that the dialed party sees is the inbound caller's caller ID. If you are dialing to a <Client>, you can set a client identifier as the callerId attribute. For instance, if you've set up a client for incoming calls and you are dialing to it, you could set the callerId attribute to client:thomas.

record. The 'record' attribute lets you specify whether the call will be recorded or not. By default, the call is not recorded. If you set the attribute to 'true' Restcomm will start recording when the two calls are bridged

Record attribute for Dial Conference

The current release of Restcomm doesn't support recording for Dial Conference. This feature will be available in the next release

Nesting. You can nest the following nouns within <Dial>: <Number>, <Client>, <Conference>

Examples. For examples of how to use the <Dial> verb see below.

```
<Response>
 <Dial>1-444-555-666</Dial>
</Response>
```

```
<Response>
 <Dial record="true">1-444-555-666</Dial>
</Response>
```

```
<Response>
 <Dial callerId="1555666777" record="true">1-444-555-666</Dial>
</Response>
```

3.9.1. Number

The <Number> noun specifies a phone number to dial. You can use multiple <Number> nouns within a <Dial> verb to simultaneously call all of them at once. The first call to pick up is connected to the current call and the rest are hung up.

Table 3.19. Number Attributes

Name	Allowed Values	Default Value
sendDigits	any digits	none
url	any url	none

sendDigits. The 'sendDigits' attribute tells RestComm to play DTMF tones when the call is answered. This is useful when dialing a phone number and an extension. RestComm will dial the number, and when the automated system picks up, send the DTMF tones to connect to the extension.

url. The 'url' attribute allows you to specify a url for a RCML document that will run on the called party's end, after he/she answers, but before the parties are connected. You can use this RCML to privately play or say information to the called party, or provide a chance to decline the phone call using <Gather> and <Hangup>. The current caller will continue to hear ringing while the RCML document executes on the other end. RCML documents executed in this manner are not allowed to contain the <Dial> verb.

Examples. For an example of how to use the <Number> noun see below.

```
<Response>
 <Dial>
  <Number sendDigits="wwww1234">1-444-555-6666</Number>
 </Dial>
</Response>
```

3.9.2. Client

The <Client> noun specifies a client identifier to dial. You can use multiple <Client> nouns within a <Dial> verb to simultaneously attempt a connection with many clients at once. The first client to accept the incoming connection is connected to the call and the other connection attempts are canceled.

Examples. For an example of how to use the <Client> none see below.

```
<Response>
 <Dial>
  <Client>thomas</Client>
 </Dial>
</Response>
```

3.9.3. Conference

The <Conference> noun allows you to connect to a named conference room and talk with the other callers who have also connected to that room. The name of the room is up to you and is namespaced to your account.

Table 3.20. Conference Attributes

Name	Allowed Values	Default Value
muted	true, false	false
beep	true, false	true
startConferenceOnEnter	true, false	true
endConferenceOnExit	true, false	false
waitUrl	RCML url, empty string	default RestComm hold music
waitMethod	GET or POST	POST
maxParticipants	positive integer <= 40	40

muted. The 'muted' attribute lets you specify whether a participant can speak in the conference. If this attribute is set to 'true', the participant will only be able to listen to people in the conference.

beep. The 'beep' attribute lets you specify whether a notification beep is played to the conference when a participant joins or leaves the conference.

startConferenceOnEnter. This attribute tells a conference to start when this participant joins the conference, if it is not already started. If this is false and the participant joins a conference that has not started, they are muted and hear background music until a participant joins where startConferenceOnEnter is true. This is useful for implementing moderated conferences.

endConferenceOnExit. If a participant has this attribute set to 'true', then when that participant leaves, the conference ends and all other participants drop out. This is useful for implementing moderated conferences that bridge two calls and allow either call leg to continue executing RCML if the other hangs up.

waitUrl. The 'waitUrl' attribute lets you specify a URL for music that plays before the conference has started. The URL may be a WAV or a RCML document that uses <Play> or <Say> for content. This defaults to a selection of Creative Commons licensed background music, but you can replace it with your own music and messages. If the 'waitUrl' responds with RCML, RestComm will only process <Play>, <Say>, and <Redirect> verbs. If you do not wish anything to play while waiting for the conference to start, specify the empty string (set 'waitUrl' to '').

waitMethod. This attribute indicates which HTTP method to use when requesting 'waitUrl'. It defaults to 'POST'. Be sure to use 'GET' if you are directly requesting static audio files such as WAV files so that RestComm properly caches the files.

maxParticipants. This attribute indicates the maximum number of participants you want to allow within a named conference room. The default maximum number of participants is 40. The value must be a positive integer less than or equal to 100.

Examples. For an example of how to use the <Conference> noun see below.

```
<Response>
 <Dial>
  <Conference>1234</Conference>
 </Dial>
</Response>
```

Uri noun deprecated

The Uri noun has been deprecated and will no longer be available in future releases. Please use the SIP noun instead.

3.9.4. Sip

The <Sip> noun specifies a SIP URI to dial. You can use multiple <Sip> nouns within a <Dial> verb to simultaneously attempt a connection with many user agents at once. The first user agent to accept the incoming connection is connected to the call and the other connection attempts are canceled.

The `Dial` verb's `Sip` noun lets you set up VoIP sessions by using SIP -- Session Initiation Protocol. With this feature, you can send a call to any SIP endpoint. Set up your RCML to use the `Sip` noun within the `Dial` verb.

Currently, only one `Sip` noun may be specified per `Dial`, and the INVITE message may be sent to only one SIP endpoint. Also, you cannot add any other nouns (eg Number, Client) in the same `Dial` as the SIP. If you want to use another noun, set up a callback on the `Dial` to use alternate methods.

Examples. For an example of how to use the <Sip> noun see below.

```
<Response>
 <Dial>
  <Sip>sip:alice@127.0.0.1:5080</Sip>
 </Dial>
</Response>
```

Authentication .

Send username and password attributes for authentication to your SIP infrastructure as attributes on the Sip noun.

Table 3.21. Request Parameters

Attribute Name	Values
username	Username for SIP authentication.
password	Password for SIP authentication

Authentication Example .

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
 <Dial>
  <Sip username="alice" password="secret">sip:alice@example.com</Sip>
 </Dial>
</Response>
```

Custom headers.

Send custom headers by appending them to the SIP URI -- just as you'd pass headers in a URI over HTTP. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
 <Dial>
  <Sip>
  sip:alice@example.com?mycustomheader=tata&myotherheader=toto
  </Sip>
 </Dial>
</Response>
```

Character Limit

While the SIP URI itself must be under 255 chars, the headers must be under 1024 characters.

Transport.

Set a parameter on your SIP URI to specify what transport protocol you want to use. Currently, this is limited to TCP and UDP. By default, Restcomm sends your SIP INVITE over UDP. Change this by using the transport parameter:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
 <Dial>
  <Sip>
  sip:alice@example.com;transport=tcp
  </Sip>
 </Dial>
</Response>
```

Attributes.

Table 3.22. Request Parameters

Attribute Name	Allowed Values	Default Value
url	call screening url.	none.
method	GET, POST	POST

The url attribute allows you to specify a url for a RCML document that runs on the called party's end, after they answer, but before the two parties are connected. You can use this RCML to privately Play or Say information to the called party, or provide a chance to decline the phone call using Gather and Hangup. The current caller continues to hear ringing while the RCML document executes on the other end. RCML documents executed in this manner cannot contain the Dial verb.

method.

The method attribute allows you to specify which HTTP method Restcomm should use when requesting the URL specified in the url attribute. The default is POST.

Call Screening HTTP parameters.

When a call is answered, Restcomm passes the following parameters with its request to your screening URL (in addition to the standard RCML Voice request parameters):

Table 3.23. Request Parameters

Attribute Name	Values
SipCallId	The SIP call ID header of the request made to the remote SIP infrastructure.
SipHeader	The name/value of any X-headers returned in the 200 response to the SIP INVITE request.

Dial Action HTTP parameters.

Use the action callback parameters to modify your application based on the results of the SIP dial attempt:

Table 3.24. Request Parameters

Attribute Name	Values
DialSipCallId	The SIP call ID header of the request made to the remote SIP infrastructure.
DialSipResponseCode	The SIP response code as a result of the INVITE attempt.
DialSipHeader_	The name/value of any X-headers returned in the final response to the SIP INVITE request.

Dial with Multiple Examples.

A more complex Dial, specifying custom settings as attributes on Dial, including call screening and setting the protocol to TCP.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
 <Dial
  record="true"
  timeout="10"
  hangupOnStar="true"
  callerId="bob"
  method="POST"
  action="/handle_post_dial">
 <Sip
  method="POST"
  url="/handle_screening_on_answer">
  sip:alice@example.com?customheader=foo
 </Sip>
 </Dial>
</Response>
```

3.10. Hangup

The <Hangup> verb ends a call.

Nesting. The <Hangup> verb can not have any other verbs or nouns nested.

Examples. For an example of how to use the <Hangup> verb see below.

```
 <Response>
  <Hangup/>
 </Response>
```

3.11. Redirect

The <Redirect> verb transfers control of a call to the RCML at a different URL. All verbs after <Redirect> are unreachable and ignored.

Table 3.25. Redirect Attributes

Name	Allowed Values	Default Value
method	GET, POST	POST

method. The 'method' attribute takes the value 'GET' or 'POST'. This tells RestComm whether to request the URL via HTTP GET or POST.

Nesting. The <Redirect> verb can not have any other verbs or nouns nested.

Examples. For an example of how to use the <Redirect> verb see below.

```
<Response>
 <Redirect>http://foobar.com/instructions</Redirect>
</Response>
```

3.12. Reject

The <Reject> verb rejects an incoming call to your RestComm endpoint. This is useful for blocking unwanted calls. If the first verb in a RCML response is <Reject>, RestComm will not pick up the call. The call ends with a status of 'busy' or 'no-answer', depending on the verb's 'reason' attribute. Any verbs after <Reject> are unreachable and ignored.

Table 3.26. Reject Attributes

Name	Allowed Values	Default Value
reason	rejected, busy	rejected

reason. The reason attribute takes the values "rejected" and "busy." This tells RestComm what message to play when rejecting a call. Selecting "busy" will play a busy signal to the caller, while selecting "rejected" will play a standard not-in-service response.

Nesting. The <Reject> verb can not have any other verbs or nouns nested.

Examples. For an example of how to use the <Reject> verb see below.

```
<Response>
 <Reject reason="busy"/>
</Response>
```

3.13. Pause

The <Pause> verb waits silently for a specific number of seconds. If <Pause> is the first verb in a RCML response, RestComm will wait the specified number of seconds before picking up the call.

Table 3.27. Pause Attributes

Name	Allowed Values	Default Value
length	integer > 0	1 second

length. The 'length' attribute specifies how many seconds RestComm will wait silently before continuing on.

Nesting. The <Pause> verb can not have any other verbs or nouns nested.

Examples. For an example of how to use the <Pause> verb see below.

```
<Response>
 <Pause length="5"/>
</Response>
```

3.14. UssdCollect

The UssdCollect verb is a USSD integration with Restcomm resource. When fully configured with a USSD gateway, Restcomm will send all UssdMessage to the configured gateway

Example. For an example of how to use the USSDCollect verb see below.

```
<Response>
 <UssdCollect action="http://my.controller.net">
  <UssdMessage> 1 for first option</UssdMessage>
  <UssdMessage> 2 for first option</UssdMessage>
  <UssdMessage> 3 for first option</UssdMessage>
 </UssdCollect>
</Response>
```

3.14.1. UssdMessage

Note

Before you can send an USSD message you must first configure the restcomm.xml with the USSD gateway information. See Section 4.13, “UssdPush” for more information.

The UssdMessage verb contains the message that Restcomm will send to the USSD gateway

Nesting. The UssdMessage can be nested inside of the USSDCollect verb.

Examples. For an example of how to use the USSDCollect verb see below.

```
<Response>
 <UssdMessage>Test USSD Message</UssdMessage></UssdCollect>
</Response>
```

3.14.2. Language

The language noun let you specify the language of the USSD message that is sent to the USSD gateway

Table 3.28. Language Attributes

Name	Allowed Values	Default Value
Language	bf, bp, en, en-gb,cf, cs, dan, fi es, fr, de, el, it, nl, no, pl, pt, ru, ar, ca, sv, th, tr	en

Nesting. The Language noun cannot be nested. If not specified, English language is default

Example. For an example of how to use the USSD Language noun see below.

```
<Response>
 <Language>fr</Language>
 <UssdMessage>Test USSD Message</UssdMessage></UssdCollect>
</Response>
```

Chapter 4. Restful APIs

The RestComm Restufl APIs are very similar to Twilio's and allow you to query meta-data about your account, phone numbers, calls, text messages, and recordings. Through the Restful API you can also start outbound calls and send text messages.

Resource Encoding. When an HTTP client makes a request to RestComm for a resource the HTTP client has the option to pick which encoding is used for the response. Currently, RestComm supports XML and JSON encoding. The default encoding is XML but to receive the resource in JSON just append .json to the end of the resource name.

User Authentication. Before accepting any Restful API call, Restcomm will authenticate the request source using either a valid Account SID or an Email Account combined with an Authentication Token. See below for the default Administrator account.

```
{AccountSID}:{AuthToken}
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166
```

or

```
{AccountEmail}:{AuthToken}
administrator@company.com:77f8c12cc7b8f8423e5c38b035249166
```

4.1. Accounts

Accounts and sub-accounts are useful for things like segmenting phone numbers and usage data for your users and controlling access to data.

Account Resource URI. /2012-04-24/Accounts/{AccountSid}

Table 4.1. Resource Properties

Property	Description
Sid	A string that uniquely identifies this account.
DateCreated	The date that this account was created.

Property	Description
DateUpdated	The date that this account was last updated.
FriendlyName	A description of this account, up to 64 characters long. By default the FriendlyName is your email address.
Status	The status of this account. Possible values are active, suspended, and closed.
AuthToken	The authorization token for this account. This should not be shared.
Uri	The URI for this account, relative to http://localhost:port/restcomm.

4.1.1. Supported Operations

HTTP GET. Returns the representation of an Account resource, including the properties above.

Account Resource URI. /2012-04-24/Accounts/{EmailAddress}

HTTP POST/PUT. Modifies an Account resource and returns the representation, including the properties above. Below you will find a list of optional parameters.

Table 4.2. Request Parameters

Parameter	Description
FriendlyName	A description of this account, up to 64 characters long.
Status	The status of this account. Possible values are active, suspended, and closed.
Password	A password that will be used to generate the AuthToken for the new Account resource.

Get information about the default account.

```
curl -X GET http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf
```

Change default account password(AuthToken).

```
curl -X PUT http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf -d "Password=NewPassword"
```

Command line versus Browser

The above command uses the Account SID and the one below uses the Email Account. Note the `administrator%40company.com` is used instead of `administrator@company.com` . This is because using curl on the bash terminal doesn't parse the @ correctlyl. If you were to running on a browser, you can safely use the @ as the web browser will correctly handle it.

```
curl -X GET http://administrator
%40company.com:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/restcomm/2012-04-24/Accounts/
ACae6e420f425248d6a26948c17a9e2acf
```

The above commands will print an output similar to the one below:

```
<RestcommResponse>
  <Account>
    <Sid>ACae6e420f425248d6a26948c17a9e2acf</Sid>
    <FriendlyName>Default Administrator Account</FriendlyName>
    <Status>active</Status>
    <Type>Full</Type>
    <DateCreated>2012-04-24T00:00:00.000-06:00</DateCreated>
    <DateUpdated>2012-04-24T00:00:00.000-06:00</DateUpdated>
    <AuthToken>77f8c12cc7b8f8423e5c38b035249166</AuthToken>
    <Uri>/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf</Uri>
    <SubresourceUris>
      <AvailablePhoneNumbers>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
AvailablePhoneNumbers</AvailablePhoneNumbers>
      <Calls>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls</Calls>
      <Conferences>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
Conferences</Conferences>
      <IncomingPhoneNumbers>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
IncomingPhoneNumbers</IncomingPhoneNumbers>
```

```
              <Notifications>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
Notifications</Notifications>
          <OutgoingCallerIds>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
OutgoingCallerIds</OutgoingCallerIds>
     <Recordings>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Recordings</
Recordings>
           <Sandbox>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Sandbox</
Sandbox>
            <SMSMessages>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/SMS/
Messages</SMSMessages>
             <Transcriptions>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
Transcriptions</Transcriptions>
    </SubresourceUris>
  </Account>
```

4.1.2. Account List Resource

Account List Resource URI. /2012-04-24/Accounts

4.1.2.1. Supported Operations

HTTP GET. Returns the list representation of all the Sub-Account resources for this Account, including the properties above.

HTTP POST. Creates a new Sub-Account and returns the representation of the Sub-Account resource, including the properties above. Below you will find a list of required and optional parameters.

Table 4.3. Request Parameters

Parameter	Description
EmailAddress(Required)	The email address to use for this account.
FriendlyName	A description of this account, up to 64 characters long. Default, is your email address.
Status	The status of this account. Default is active, possible values are active, suspended, and closed.
Password(Required)	A password that will be used to generate the AuthToken for the new Account resource.
Role(Required)	The security role that this Account resource will use. If no role is provided then the role of the account resource creating this will be inherited to the new Account resource and may compromise the system.

Here is an example of how to createa a sub-account. The sub-account will inherit the same permissions has the Administrator's account.

```
curl -X POST http://administrator
%40company.com:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/restcomm/2012-04-24/Accounts/ -
d "FriendlyName=MySubAccount" -d "EmailAddress=test@telestax.com" -d "Password=restcomm"
```

About Sub-accounts

Note that the SID, Email and the AuthToken (see output below) of the sub-account can now be used instead of the Administrator's account

```
<RestcommResponse>
  <Account>
    <Sid>AC3b8f0dd2e5026abde018446cbb3b185d</Sid>
    <FriendlyName>MySubAccount</FriendlyName>
    <Status>active</Status>
    <Type>Full</Type>
    <DateCreated>2013-10-16T09:22:28.708-06:00</DateCreated>
    <DateUpdated>2013-10-16T09:22:28.712-06:00</DateUpdated>
    <AuthToken>53134d7a9914e2b47c8435ebdb50ded3</AuthToken>
    <Uri>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d</Uri>
    <SubresourceUris>
      <AvailablePhoneNumbers>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/
AvailablePhoneNumbers</AvailablePhoneNumbers>
      <Calls>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/Calls</Calls>
      <Conferences>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/
Conferences</Conferences>
      <IncomingPhoneNumbers>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/
IncomingPhoneNumbers</IncomingPhoneNumbers>
      <Notifications>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/
Notifications</Notifications>
      <OutgoingCallerIds>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/
OutgoingCallerIds</OutgoingCallerIds>
      <Recordings>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/Recordings</
Recordings>
      <Sandbox>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/Sandbox</
Sandbox>
      <SMSMessages>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/SMS/
Messages</SMSMessages>
      <Transcriptions>/restcomm/2012-04-24/Accounts/AC3b8f0dd2e5026abde018446cbb3b185d/
Transcriptions</Transcriptions>
    </SubresourceUris>
```

```
</Account>
```

4.2. AvailablePhoneNumbers

The AvailablePhoneNumbers subresources let you search for incoming local and toll-free phone numbers that are available for you to purchase from a TeleStax partner.

AvailablePhoneNumbers List Resource URI. /2012-04-24/Accounts/{AccountSid}/AvailablePhoneNumbers/US/Local

Searching For Numbers. When using RestComm the way to search for new phone numbers is by searching the AvailablePhoneNumbers list resource and providing the desired area code as a filter.

Table 4.4. Resource Properties

Property	Description
FriendlyName	A friendly version of the phone number.
PhoneNumber	The phone number, in E.164 format.
Lata	The LATA for this phone number.
RateCenter	The rate center for this phone number.
Latitude	The latitude coordinate for this phone number.
Longitude	The longitude coordinate for this phone number.
Region	The two-letter state or province abbreviation for this phone number.
PostalCode	The zip code for this phone number.
IsoCountry	The ISO country code for this phone number..

4.2.1. Supported Operations

HTTP GET. Returns the representation of an AvailablePhoneNumber resource, including the properties above.

Table 4.5. Request Parameters

Property	Description
AreaCode	A three digit area code inside the U.S.

Querying Available Phone Numbers

You need to be using RestComm for VoIP Innovations in order to be able to use this feature. See here for more details RestComm AMI for VoIP Innovation [www.telestax.com/online-documentation]

Here is an example, the AreaCode is any valid United States Code

```
curl -G http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@<Elastic_IP>:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/AvailablePhoneNumbers/US/Local
 -d "AreaCode=305"
```

4.3. Gateways

The Gateways subresources let you create sip accounts that Restcomm will use to register itself to the Gateway and receive incoming traffic.

Gatewat List Resource URI. /2012-04-24/Accounts/{AccountSid}/Management/Gateways

Register Restcomm instance to a SIP Gateway. You might need to register Restcomm instnace to a SIP Gateway and receive incoming traffic from that gateway. For that you need to use Gateway REST endpoint

Table 4.6. Resource Properties

Property	Description
FriendlyName	A friendly version of the gateway.
UserName	The username that will be used to register to this gateway
Password	The password that will be used to register to this gateway
Proxy	The proxy address of the gateway
Register	Boolean flag to register or not the gateway
TTL	Time to live for the Register

4.3.1. Supported Operations

HTTP GET. Returns the representation of a Gateway resource, including the properties above. Resource URI: /2010-04-01/Accounts/{AccountSid}/Management/Gateways/{GatewaySid}

HTTP POST. Creates a new Gateway resource and returns the representation of the resource, including the properties above. Resource URI: /2010-04-01/Accounts/{AccountSid}/Management/Gateways

HTTP POST/PUT. Update a Gateway resource and returns the representation of the resource, including the properties above. Resource URI: /2010-04-01/Accounts/{AccountSid}/Management/Gateways/{GatewaySid}

HTTP DELETE. Deletes a Gateway resource. Resource URI: /2010-04-01/Accounts/{AccountSid}/Management/Gateways/{GatewaySid}

4.3.2. Examples

Create a new Gateway.

```
curl                          -X                          POST                          http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@<RESTCOMM_IP>:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Management/Gateways         -d
     "FriendlyName=mygateway"     -d     "UserName=username"     -d     "Password=password"     -d
 "Proxy=my.gateway.com" -d "Register=true" -d "TTL=3600"
```

Get a list of available Gateways.

```
curl                                        -G                                        http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@<RESTCOMM_IP>:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Management/Gateways
```

Update an existing Gateway.

```
curl -X POST http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@<RESTCOMM_IP>:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Management/Gateways/
GW106bc6f34bd24790a435eaeccc1aed72 -d "FriendlyName=MyGatewayNewName" -d "UserName=newUserName"
```

Delete an existing Gateway.

```
curl -X DELETE http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@<RESTCOMM_IP>:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Management/Gateways/
GW1cffb069192a45f2b5f5af2e76489550
```

4.4. Clients

An Client instance resource represents a user agent registered with RestComm.

Client Resource URI. /2012-04-24/Accounts/{AccountSid}/Clients/{ClientSid}

Using SIP User Agents. When using RestComm to handle SIP user agent you have to create a new Client resource, this resource acts as an account for your user agent and also dictates how calls made by the user agent should be handled.

Client without VoiceUrl

Restcomm has a new implied behavior when VoiceUrl is not provided for a Client account. Restcomm will proxy calls from such Clients to the destination Client (only if registered) or to the destination Application DID.

Only registered Clients are allowed to use the B2BUA/P2P/Proxy functionalities of Restcomm. Proxying and P2P calls are only allowed between registered(authenticated) Clients.

Table 4.7. Resource Properties

Property	Description
Sid	A string that uniquely identifies this client.
DateCreated	The date that this client was created.

Property	Description
DateUpdated	The date that this clientr was last updated.
FriendlyName	A friendly name for this client.
AccountSid	The unique id of the Account that owns this phone number.
ApiVersion	Calls to this phone number will create a new RCML session with this API version.
Login	The name that is used inside the <Client> noun. This is also used by the user agent as the user name used for registration and outbound dialing.
Password	The password used by the user agent during registration and outbound dialing.
Status	The client status the possible values are 0 for disabled and 1 for enabled.
VoiceUrl	The URL RestComm will request when this client makes an outbound call.
VoiceMethod	The HTTP method RestComm will use when requesting the above Url. Either GET or POST.
VoiceFallbackUrl	The URL that RestComm will request if execution of VoiceUrl fails for any reason.
VoiceFallbackMethod	The HTTP method RestComm will use when requesting the VoiceFallbackUrl. Either GET or POST.
VoiceApplicationSid	If this entry contains an Sid to a voice application then RestComm will ignore these voice URLs and use the voice URLs specified by the voice application.
StatusCallback	The URL that RestComm will request to pass status parameters (such as the call state) to your application.
StatusCallbackMethod	The HTTP method RestComm will use to make requests to the StatusCallback URL. Either GET or POST.
Uri	The URI for this Client, relative to http://localhost:port/restcomm.

4.4.1. Supported Operations

HTTP GET. Returns the representation of an Client resource, including the properties above.

HTTP POST/PUT. Modifies a Client resource and returns the representation, including the properties above. Below you will find a list of optional parameters.

Table 4.8. Request Parameters

Parameter	Description
FriendlyName	A formatted version of this phone number.
Password	The password used by the user agent during registration and outbound dialing.
Status	The client status the possible values are 0 for disabled and 1 for enabled.
VoiceUrl	The URL RestComm will request when this phone number receives a call.
VoiceMethod	The HTTP method RestComm will use when requesting the above Url. Either GET or POST.
VoiceFallbackUrl	The URL that RestComm will request if execution of VoiceUrl fails for any reason.
VoiceFallbackMethod	The HTTP method RestComm will use when requesting the VoiceFallbackUrl. Either GET or POST.
VoiceApplicationSid	If this entry contains an Sid to a voice application then RestComm will ignore these voice URLs and use the voice URLs specified by the voice application.
StatusCallback	The URL that RestComm will request to pass status parameters (such as the call state) to your application.
StatusCallbackMethod	The HTTP method RestComm will use to make requests to the StatusCallback URL. Either GET or POST.

HTTP DELETE. Deletes a Client from the user's Account.

4.4.2. Client List Resource

Client List Resource URI. /2012-04-24/Accounts/{AccountSid}/Clients

4.4.2.1. Supported Operations

HTTP GET. Returns the list representation of all the Client resources for this Account, including the properties above.

HTTP POST. Creates a new Client and returns the representation of the resource, including the properties above. Below you will find a list of required and optional parameters.

Table 4.9. Request Parameters

Parameter	Description
FriendlyName	A formatted version of this phone number.
Login	The name that is used inside the <Client> noun. This is also used by the user agent as the user name used for registration and outbound dialing.
Password	The password used by the user agent during registration and outbound dialing.
Status	The client status the possible values are 0 for disabled and 1 for enabled.
VoiceUrl	The URL RestComm will request when this phone number receives a call.
VoiceMethod	The HTTP method RestComm will use when requesting the above Url. Either GET or POST.
VoiceFallbackUrl	The URL that RestComm will request if execution of VoiceUrl fails for any reason.
VoiceFallbackMethod	The HTTP method RestComm will use when requesting the VoiceFallbackUrl. Either GET or POST.
VoiceApplicationSid	If this entry contains an Sid to a voice application then RestComm will ignore these voice URLs and use the voice URLs specified by the voice application.
StatusCallback	The URL that RestComm will request to pass status parameters (such as the call state) to your application.
StatusCallbackMethod	The HTTP method RestComm will use to make requests to the StatusCallback URL. Either GET or POST.

Create a Client. The client name will be Alice as shown below

```
                curl                -X                POST                                 http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Clients.json -d "Login=alice" -
d "Password=test"
```

The output of the command will be similar to the one below

```
{
  "sid": "CL4e10e3b56a614414bcc1eeca5d96effe",
  "date_created": "2013-10-16T08:51:32.460-06:00",
  "date_updated": "2013-10-16T08:51:32.460-06:00",
  "account_sid": "ACae6e420f425248d6a26948c17a9e2acf",
  "api_version": "2012-04-24",
  "friendly_name": "alice",
  "login": "alice",
  "password": "test",
  "status": "1",
  "voice_method": "POST",
  "voice_fallback_method": "POST",
          "uri":     "/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Clients/
CL4e10e3b56a614414bcc1eeca5d96effe.json"
```

Delete a Client. You must use the Client SID

```
curl                         -X                          DELETE                            http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Clients/
CL4e10e3b56a614414bcc1eeca5d96effe
```

Change Client's Password. You must use the Client SID as shown below:

```
curl                          -X                          PUT                              http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
```

```
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Clients/
CL4e10e3b56a614414bcc1eeca5d96effe -d "Password=NewPassword"
```

Get List of available Clients. The command below shows all Clients created using the default Admin Account

```
curl                        -X                          GET                          http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Clients/
```

4.5. IncomingPhoneNumbers

An IncomingPhoneNumber instance resource represents a RestComm phone number.

IncomingPhoneNumber Resource URI. /2012-04-24/Accounts/{AccountSid}/IncomingPhoneNumbers/{IncomingPhoneNumberSid}

Binding a Phone Number to an Application. When using RestComm the way to bind a phone number to an application is be creating a new IncomingPhoneNumber resource and providing the VoiceUrl to your application.

Table 4.10. Resource Properties

Property	Description
Sid	A string that uniquely identifies this incoming phone number.
DateCreated	The date that this incoming phone number was created.
DateUpdated	The date that this incoming phone number was last updated.
FriendlyName	A formatted version of this phone number.
AccountSid	The unique id of the Account that owns this phone number.
PhoneNumber	The incoming phone number in E.164 format ex. +2223334444
ApiVersion	Calls to this phone number will create a new RCML session with this API version.

Property	Description
VoiceCallerIdLookup	Look up the caller's caller-ID name. Either true or false.
VoiceUrl	The URL RestComm will request when this phone number receives a call.
VoiceMethod	The HTTP method RestComm will use when requesting the above Url. Either GET or POST.
VoiceFallbackUrl	The URL that RestComm will request if execution of VoiceUrl fails for any reason.
VoiceFallbackMethod	The HTTP method RestComm will use when requesting the VoiceFallbackUrl. Either GET or POST.
StatusCallback	The URL that RestComm will request to pass status parameters (such as the call state) to your application.
StatusCallbackMethod	The HTTP method RestComm will use to make requests to the StatusCallback URL. Either GET or POST.
SmsUrl	The URL that RestComm will request when receiving an incoming SMS message to this number. This may not be supported. Please consult with your DID provider.
SmsMethod	The HTTP method RestComm will use when making requests to the SmsUrl. Either GET or POST.
SmsFallbackUrl	The URL that RestComm will request if SmsUrl fail for any reason. Please see SmsUrl as this feature may not be supported.
SmsFallbackMethod	The HTTP method RestComm will use when making requests to SmsFallbackUrl. Either GET or POST.
Uri	The URI for this incoming phone number, relative to http://localhost:port/restcomm.

4.5.1. Supported Operations

HTTP GET. Returns the representation of an IncomingPhoneNumber resource, including the properties above.

HTTP POST/PUT. Modifies an IncomingPhoneNumber resource and returns the representation, including the properties above. Below you will find a list of optional parameters.

Table 4.11. Request Parameters

Parameter	Description
FriendlyName	A formatted version of this phone number.
ApiVersion	Calls to this phone number will create a new RCML session with this API version.
VoiceCallerIdLookup	Look up the caller's caller-ID name. Either true or false.
VoiceUrl	The URL RestComm will request when this phone number receives a call.
VoiceMethod	The HTTP method RestComm will use when requesting the above Url. Either GET or POST.
VoiceFallbackUrl	The URL that RestComm will request if execution of VoiceUrl fails for any reason.
VoiceFallbackMethod	The HTTP method RestComm will use when requesting the VoiceFallbackUrl. Either GET or POST.
StatusCallback	The URL that RestComm will request to pass status parameters (such as the call state) to your application.
StatusCallbackMethod	The HTTP method RestComm will use to make requests to the StatusCallback URL. Either GET or POST.
SmsUrl	The URL that RestComm will request when receiving an incoming SMS message to this number. This may not be supported. Please consult with your DID provider.
SmsMethod	The HTTP method RestComm will use when making requests to the SmsUrl. Either GET or POST.
SmsFallbackUrl	The URL that RestComm will request if SmsUrl fail for any reason. Please see SmsUrl as this feature may not be supported.
SmsFallbackMethod	The HTTP method RestComm will use when making requests to SmsFallbackUrl. Either GET or POST.

HTTP DELETE. Releases an IncomingPhoneNumber from the user's Account.

4.5.2. IncomingPhoneNumber List Resource

IncomingPhoneNumber List Resource URI. /2012-04-24/Accounts/{AccountSid}/IncomingPhoneNumbers

4.5.2.1. Supported Operations

HTTP GET. Returns the list representation of all the IncomingPhoneNumber resources for this Account, including the properties above.

HTTP POST. Creates a new IncomingPhoneNumber and returns the representation of the resource, including the properties above. Below you will find a list of required and optional parameters.

Table 4.12. Request Parameters

Parameter	Description
PhoneNumber(Required)	The phone number you want to provision.
FriendlyName	A formatted version of this phone number.
ApiVersion	Calls to this phone number will create a new RCML session with this API version.
VoiceCallerIdLookup	Look up the caller's caller-ID name. Either true or false.
VoiceUrl	The URL RestComm will request when this phone number receives a call.
VoiceMethod	The HTTP method RestComm will use when requesting the above Url. Either GET or POST.
VoiceFallbackUrl	The URL that RestComm will request if execution of VoiceUrl fails for any reason.
VoiceFallbackMethod	The HTTP method RestComm will use when requesting the VoiceFallbackUrl. Either GET or POST.
StatusCallback	The URL that RestComm will request to pass status parameters (such as the call state) to your application.
StatusCallbackMethod	The HTTP method RestComm will use to make requests to the StatusCallback URL. Either GET or POST.
SmsUrl	The URL that RestComm will request when receiving an incoming SMS message to this number. This may not be supported. Please consult with your DID provider.

Parameter	Description
SmsMethod	The HTTP method RestComm will use when making requests to the SmsUrl. Either GET or POST.
SmsFallbackUrl	The URL that RestComm will request if SmsUrl fail for any reason. Please see SmsUrl as this feature may not be supported.
SmsFallbackMethod	The HTTP method RestComm will use when making requests to SmsFallbackUrl. Either GET or POST.

Attach a phone number to an application. This one uses the default application

```
curl                -X                POST                                 http://administrator
%40company.com:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/restcomm/2012-04-24/Accounts/
ACae6e420f425248d6a26948c17a9e2acf/IncomingPhoneNumbers.json    -d    "PhoneNumber=1234"    -d
 "VoiceUrl=http://127.0.0.1:8080/restcomm/demos/hello-play.xml"
```

```
Result of the above command

{
  "sid": "PNdd7a0a0248244615978bd5781598e5eb",
  "account_sid": "ACae6e420f425248d6a26948c17a9e2acf",
  "friendly_name": "234",
  "phone_number": "+1234",
  "voice_url": "http://127.0.0.1:8080/restcomm/demos/hello-play.xml",
  "voice_method": "POST",
  "voice_fallback_method": "POST",
  "status_callback_method": "POST",
  "voice_caller_id_lookup": false,
  "date_created": "2013-10-04T17:42:02.500-06:00",
  "date_updated": "2013-10-04T17:42:02.500-06:00",
  "sms_method": "POST",
  "sms_fallback_method": "POST",
  "api_version": "2012-04-24",
               "uri":       "/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
IncomingPhoneNumbers/PNdd7a0a0248244615978bd5781598e5eb.json"
```

Delete a phone number. You have to get the SID of the phone and use curl to delete as follows

```
curl                    -X                    DELETE                    http://administrator
%40company.com:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/restcomm/2012-04-24/Accounts/
ACae6e420f425248d6a26948c17a9e2acf/IncomingPhoneNumbers/PNdd7a0a0248244615978bd5781598e5eb
```

List of phone numbers. Gets all numbers created using IncomingPhoneNumbers.json

```
curl                -X                GET                                   http://administrator
%40company.com:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/restcomm/2012-04-24/Accounts/
ACae6e420f425248d6a26948c17a9e2acf/IncomingPhoneNumbers.json
```

4.6. Calls

A Call represents a connection between a phone or user agent and RestComm. This may be inbound or outbound. The Calls list resource represents the set of phone calls originated and terminated from an account.

Call Resource URI. /2012-04-24/Accounts/{AccountSid}/Calls/{CallSid}

Table 4.13. Resource Properties

Property	Description
Sid	A string that uniquely identifies this call.
ParentCallSid	A string that uniquely identifies the call that created this leg.
DateCreated	The date that this call was created.
DateUpdated	The date that this call was last updated.
AccountSid	The unique id of the Account that created this call.
To	The phone number or identifier that will be the recipient of this call.
From	The phone number or identifier that originated this call.
PhoneNumberSid	If the call was inbound, this is the Sid of the IncomingPhoneNumber that received the call.
Status	A string representing the status of the call. Possible values are queued, ringing, in-

Property	Description
	progress, completed, failed, busy and no-answer.
StartTime	The start time of the call. Empty if the call has not yet been started.
EndTime	The end time of the call. Empty if the call has not ended..
Duration	The length of the call in seconds.
Price	The charge for this call, in the currency associated with the account. Populated after the call is completed.
Direction	A string describing the direction of the call. Possible values are inbound, outbound-api, and outbound-dial
AnsweredBy	If this call was initiated with answering machine detection, either human or machine. Empty otherwise.
ApiVersion	Displays the current API version
ForwardFrom	If this call was an incoming call forwarded from another number, the forwarding phone number (depends on carrier supporting forwarding). Empty otherwise.
CallerName	If this call was an incoming call, the caller's name. Empty otherwise.
Uri	The URI for this account, relative to http://localhost:port/restcomm.

4.6.1. Supported Operations

HTTP GET. Returns the representation of a Call resource, including the properties above.

4.6.2. Call List Resource

Call List Resource URI. /2012-04-24/Accounts/{AccountSid}/Calls

4.6.3. Supported Operations

HTTP GET. Returns the list representation of all the Call resources for this Account, including the properties above.

HTTP POST. Makes a new Call and returns the representation of the Call resource, including the properties above. Below you will find a list of required and optional parameters.

Table 4.14. Request Parameters

Parameter	Description
From(Required)	The phone number to use as the caller id.
To(Required)	The phone number to call.
Url(Required)	The fully qualified URL that should be executed when the call connects.
Method	The HTTP method RestComm should use when making its request to the above Url. Defaults to POST.
FallbackUrl	The URL that RestComm will request if execution of Url fails for any reason.
FallbackMethod	The HTTP method that RestComm should use to request the FallbackUrl. Must be either GET or POST. Defaults to POST.
StatusCallback	A URL that RestComm will request when the call ends to notify your app.
StatusCallbackMethod	The HTTP method RestComm should use when requesting the above StatusCallback. Defaults to POST.
Timeout	The number of seconds that RestComm should allow the phone to ring before assuming there is no answer. The default is 60 seconds.

Making a call to a SIP account.

Restcomm will make a call to any SIP account that is reachable. It the example below, the SIP account is listening on port 5060. When you make the call, the SIP phone on which Alice is registered will ring and the hello-play.xml file will be played.

```
curl -X POST http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls.json -d "From=
+16175551212" -d "To=sip:alice@127.0.0.1:5060" -d "Url=http://127.0.0.1:8080/restcomm/demos/
hello-play.xml"
```

Making a call to a Restcomm client.

You must first create a RestComm client. In the example below, the Restcomm client created is called Alice. When you make the call, the SIP phone on which Alice is registered will ring and the hello-play.xml file will be played.

```
curl -X POST http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls.json -d "From=
+16175551212" -d "To=client:alice" -d "Url=http://127.0.0.1:8080/restcomm/demos/hello-play.xml"
```

Calling a DID number

The above example shows how to make a call to a SIP number. If you want to make a call to a DID number, you must can connect Restcomm to a DID provisioning service provider. The quickest way is to use RestComm AMI for Voice Innovation [http://www.telestax.com/voip-innovations-offers-restcomm/]

Get a list of all available calls. This will return all the available calls linked to the account SID

Working on a production server

Using filter is a good practice on a server with thousands or millions of calls

```
curl -X GET http://administrator
%40company.com:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/restcomm/2012-04-24/Accounts/
ACae6e420f425248d6a26948c17a9e2acf/Calls
```

If the system hasn't received any calls, you will see the the output below

```
<RestcommResponse>
                <Calls page="0" numpages="0" pagesize="50" total="0"
    start="0" end="0" uri="/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
Calls" firstpageuri="/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls?
Page=0&amp;PageSize=50" previouspageuri="null" nextpageuri="null" lastpageuri=="/
```

```
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls?Page=0&amp;PageSize=50"/
>
```

4.6.4. REST API: Modifying Live Calls

Realtime call modification allows you to interrupt an in-progress call and terminate it or have it begin processing RCML from a new URL. This is useful for any application where you want to asynchronously change the behavior of a running call. For example: hold music, call queues, transferring calls, forcing hangup, etc.

4.6.4.1. HTTP POST to a Call

Client List Resource URI.

To redirect or terminate a live call, you make an HTTP POST request to an in-progress Call instance resource URI:

/2010-04-01/Accounts/{AccountSid}/Calls/{CallSid}

POST Parameters.

The following parameters are available for you to POST when modifying a phone call:

Table 4.15. Request Parameters

Parameter	Description
Url	A valid URL that returns TwiML. Twilio will immediately redirect the call to the new TwiML.
Method	The HTTP method Twilio should use when requesting the above URL. Defaults to POST.
Status	Either canceled or completed. Specifying canceled will attempt to hangup calls that are queued or ringing but not affect calls already in progress. Specifying completed will attempt to hang up a call even if it's already in progress.

> **Call in-Progress**
>
> Note that any call which is currently ringing within a Dial verb is in-progress from the point of view of Restcomm, and thus you must use 'Status=completed' to cancel it.

Optional Parameters. You may POST the following parameters:

Table 4.16. Request Parameters

Parameter	Description
FallbackUrl	A URL that Twilio will request if an error occurs requesting or executing the TwiML at Url.
FallbackMethod	The HTTP method that Twilio should use to request the FallbackUrl. Must be either GET or POST. Defaults to POST.
StatusCallback	A URL that Twilio will request when the call ends to notify your app.
StatusCallbackMethod	The HTTP method Twilio should use when requesting the above URL. Defaults to POST.

4.6.4.2. Modifying Live Calls - Example

```
 Modifying a Live Call

In order to accomplish this, you need to create a client called alice
Start a SIP phone and register alice
From the terminal run the following curl command
Make sure alice is using the port 5061
The "From=" could be any number of your choice
The Url is the default sample example provided with Restcomm

curl                          -X                          POST                          http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls.json      -d      "From=
+16175551212"  -d  "To=sip:alice@127.0.0.1:5061"  -d  "Url=http://127.0.0.1:8080/restcomm/demos/
hello-play.xml"

You will see an output similar to the one below:

{
  "sid": "CAfa51b104354440b09213d04752f50271",
  "date_created": "2013-11-01T03:41:14.488-06:00",
  "date_updated": "2013-11-01T03:41:14.488-06:00",
  "account_sid": "ACae6e420f425248d6a26948c17a9e2acf",
  "to": "alice",
  "from": "+16175551212",
  "status": "queued",
  "start_time": "2013-11-01T03:41:14.488-06:00",
  "price": "0.0",
  "direction": "outbound-api",
  "api_version": "2012-04-24",
                 "uri":         "/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CAfa51b104354440b09213d04752f50271.json",
  "subresource_uris": {
     "notifications":  "/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CAfa51b104354440b09213d04752f50271/Notifications",
       "recordings":  "/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CAfa51b104354440b09213d04752f50271/Recordings"
  }


Notice the "sid": "CAfa51b104354440b09213d04752f50271",
This Call ID is what you must use to interact with the current call.
You can now redirect the current call to another application as shown below
Notice that the Call ID is referenced
The call will now be redirected to the Url specified(hello-world.xml)


curl                         -X                         POST                         http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CAfa51b104354440b09213d04752f50271     -d     "Url=http://127.0.0.1:8080/restcomm/demos/hello-
world.xml"

The output showing the same Call ID

<RestcommResponse>
  <Call>
    <Sid>CAfa51b104354440b09213d04752f50271</Sid>
    <DateCreated>2013-11-01T03:41:14.488-06:00</DateCreated>
    <DateUpdated>2013-11-01T03:41:14.488-06:00</DateUpdated>
    <ParentCallSid/>
    <AccountSid>ACae6e420f425248d6a26948c17a9e2acf</AccountSid>
    <To>alice</To>
    <From>+16175551212</From>
    <PhoneNumberSid/>
       ..... TRUNCATED


You can still redirect the current call back to the previous application


curl    -X    POST    http://ACae6e420f425248d6f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CAfa51b104354440b09213d04752f50271 -d "Url=http://127.0.0.1:8080/restcomm/demos/hello-play.xml"

The output showing the same Call ID

<RestcommResponse>
  <Call>
    <Sid>CAfa51b104354440b09213d04752f50271</Sid>
    <DateCreated>2013-11-01T03:41:14.488-06:00</DateCreated>
    <DateUpdated>2013-11-01T03:41:14.488-06:00</DateUpdated>
    <ParentCallSid/>
    <AccountSid>ACae6e420f425248d6a26948c17a9e2acf</AccountSid>
    <To>alice</To>
    <From>+16175551212</From>
    <PhoneNumberSid/>
   ..... TRUNCATED


You can end the call using the Status=completed command as shown below
```

```
curl     -X     POST     http://ACae6e420f425248d6f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CAfa51b104354440b09213d04752f50271 -d "Status=completed"

The output showing the same Call ID

<RestcommResponse>
  <Call>
    <Sid>CAfa51b104354440b09213d04752f50271</Sid>
    <DateCreated>2013-11-01T03:41:14.488-06:00</DateCreated>
    <DateUpdated>2013-11-01T03:41:14.488-06:00</DateUpdated>
    <ParentCallSid/>
    <AccountSid>ACae6e420f425248d6a26948c17a9e2acf</AccountSid>
    <To>alice</To>
    <From>+16175551212</From>
    <PhoneNumberSid/>
   ..... TRUNCATED
```

4.6.5. List Filter

HTTP GET. The following GET query string parameters allow you to limit the list returned. Note, parameters are case-sensitive:

Table 4.17. Request Parameters

Parameter	Description
To	Only show calls to this phone number or Client identifier.
From	Only show calls from this phone number or Client identifier.
Status	Only show calls currently in this status. May be queued, ringing, in-progress, canceled, completed, failed, busy, or no-answer.
StartTime	Only show calls that started on this date, given as YYYY-MM-DD. Also supports inequalities, such as StartTime=YYYY-MM-DD for calls that started at or before midnight on a date, and StartTime=YYYY-MM-DD for calls that started at or after midnight on a date.
ParentCallSid	Only show calls spawned by the call with this Sid.

Filter using the From parameter. The example below will only return Calls made from client Alice

```
    curl -X GET http://administrator%40company.com:77f8c12cc7166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls?From=alice
```

The result will be similar to the one below

```
<RestcommResponse>
                <Calls page="0" numpages="0" pagesize="50" total="0"
    start="0" end="1" uri="/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
Calls" firstpageuri="/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls?
Page=0&amp;PageSize=50" previouspageuri="null" nextpageuri="null" lastpageuri=="/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls?Page=0&amp;PageSize=50">
    <Call>
      <Sid>CAc0fb839632cf444f9066876d5de741e0</Sid>
      <DateCreated>2013-10-18T04:51:47.643-06:00</DateCreated>
      <DateUpdated>2013-10-18T04:51:49.174-06:00</DateUpdated>
      <ParentCallSid/>
      <AccountSid>ACae6e420f425248d6a26948c17a9e2acf</AccountSid>
      <To>1234</To>
      <From>alice</From>
      <PhoneNumberSid/>
      <Status>completed</Status>
      <StartTime>2013-10-18T04:51:47.671-06:00</StartTime>
      <EndTime>2013-10-18T04:51:49.174-06:00</EndTime>
      <Duration>1</Duration>
      <Price>0.00</Price>
      <Direction>inbound</Direction>
      <AnsweredBy/>
      <ApiVersion>2012-04-24</ApiVersion>
      <ForwardedFrom/>
      <CallerName/>
                            <Uri>/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CAc0fb839632cf444f9066876d5de741e0</Uri>
      <SubresourceUris>
        <Notifications>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CAc0fb839632cf444f9066876d5de741e0/Notifications</Notifications>
           <Recordings>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CAc0fb839632cf444f9066876d5de741e0/Recordings</Recordings>
      </SubresourceUris>
    </Call>
  </Calls>
```

4.6.6. Paging Information

HTTP GET. The following GET query string parameters allow you to limit the list returned. Note, parameters are case-sensitive:

Table 4.18. Request Parameters

Parameter	Description
Page	The current page number. Zero-indexed, so the first page is 0.
NumPages	The total number of pages.
PageSize	How many items are in each page
Total	The total number of items in the list.
Start	The position in the overall list of the first item in this page.
End	The position in the overall list of the last item in this page.

Example. The command below will return a single item from the list of calls using the PageSize parameter

```
curl -X GET http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls?PageSize=1
```

The result of the PageSize parameter

```
<RestcommResponse>
                <Calls page="0" numpages="7" pagesize="1" total="7"
    start="0" end="0" uri="/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
Calls" firstpageuri="/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls?
Page=0&amp;PageSize=1" previouspageuri="null" nextpageuri="/restcomm/2012-04-24/Accounts/
ACae6e420f425248d6a26948c17a9e2acf/Calls?
Page=1&amp;PageSize=1&amp;AfterSid=CA4049cf008d6b4277b92ab863fd4ec7c8" lastpageuri=="/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls?Page=7&amp;PageSize=1">
    <Call>
      <Sid>CA4049cf008d6b4277b92ab863fd4ec7c8</Sid>
      <DateCreated>2013-10-18T04:49:45.942-06:00</DateCreated>
      <DateUpdated>2013-10-18T04:49:46.272-06:00</DateUpdated>
      <ParentCallSid/>
      <AccountSid>ACae6e420f425248d6a26948c17a9e2acf</AccountSid>
      <To>1235</To>
      <From>bob</From>
      <PhoneNumberSid/>
      <Status>completed</Status>
      <StartTime>2013-10-18T04:49:45.994-06:00</StartTime>
      <EndTime>2013-10-18T04:49:46.272-06:00</EndTime>
      <Duration>0</Duration>
      <Price>0.00</Price>
      <Direction>inbound</Direction>
      <AnsweredBy/>
      <ApiVersion>2012-04-24</ApiVersion>
      <ForwardedFrom/>
      <CallerName/>
                             <Uri>/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CA4049cf008d6b4277b92ab863fd4ec7c8</Uri>
      <SubresourceUris>
        <Notifications>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CA4049cf008d6b4277b92ab863fd4ec7c8/Notifications</Notifications>
           <Recordings>/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Calls/
CA4049cf008d6b4277b92ab863fd4ec7c8/Recordings</Recordings>
      </SubresourceUris>
    </Call>
  </Calls>
```

Additional Paging Information. The API returns URIs to the next, previous, first and last pages of the returned list as shown in the table below:

Table 4.19. Request Parameters

Parameter	Description
Uri	The URI of the current page.
Firstpageuri	The URI for the first page of this list.
Nextpageuri	The URI for the next page of this list.
Previouspageuri	The URI for the previous page of this list.
Lastpageuri	The URI for the last page of this list.

4.7. SMS Messages

An SMS Message resource represents an inbound or outbound SMS message.

SMS Message Resource URI. /2012-04-24/Accounts/{AccountSid}/SMS/Messages/{SMSMessageSid}

Table 4.20. Resource Properties

Property	Description
Sid	A string that uniquely identifies this SMS Message.
DateCreated	The date that this SMS Message was created.
DateUpdated	The date that this SMS Message was last updated.
DateSent	The date that the SMS was sent or received by RestComm.
AccountSid	The unique id of the Account that sent or received this SMS message.
From	The phone number or short code that initiated the message.
To	The phone number or short code that received the message.
Body	The text body of the SMS message. Up to 160 characters long.
Status	The status of this SMS message. Possible values are queued, sending, sent, failed, and received.
Direction	The direction of this SMS message. Possible values are incoming, outbound-api, outbound-call.
ApiVersion	The API version RestComm used to handle the SMS message.
Uri	The URI for this account, relative to http://localhost:port/restcomm.

4.7.1. Supported Operations

HTTP GET. Returns the representation of an SMS Message resource, including the properties above.

4.7.2. SMS Message List Resource

SMS Message List Resource URI. /2012-04-24/Accounts/{AccountSid}/SMS/Messages

4.7.2.1. Supported Operations

HTTP GET. Returns the list representation of all the Call resources for this Account, including the properties above.

HTTP POST. Sends a new SMS Message and returns the representation of the SMS Message resource, including the properties above. Below you will find a list of required and optional parameters.

Table 4.21. Request Parameters

Parameter	Description
From(Required)	A phone number that is enabled for SMS.
To(Required)	The destination phone number in E.164 format.
Body(Required)	The text of the message you want to send, limited to 160 characters.
Custom headers X- (optional)	Optionally you can provide as many custom headers as you wish. The custom headers must start with X-, for example "X-MyCustom-Header=My custom header value"

Using custom headers

These additional headers will be part of the SIP MESSAGE that Restcomm will create and dispatch to the SMS Aggregator.

Using SMS and making DID calls

You need to configure Restcomm to send SMS messages and DID phone calls to a Service Provider for provisioning. In the `restcomm.xml` file, the outbound-proxy-uri and the SMS outbound-endpoint must point to the Service Provider IP address. You may also decide to use Restcomm AMI. [http://www.telestax.com/docs/]

Send SMS Messages.

Note the encoding used `%2B13216549878` instead of the `+13216549878` The + sign is encoded to to send SMS from the command line.

"From" DID number

The "From" number should be the DID SMS enabled number from VoIP Innovations.

From the bash terminal, you can run the command below:

```
curl -X POST http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/SMS/Messages -d "To=
%2B13216549878" -d "From=%2B19876543212" -d "Body=This is a test from RestComm"
```

To send the same SMS but this time also provide some additional headers:

```
curl -X POST http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/SMS/Messages -d "To=
%2B13216549878" -d "From=%2B19876543212" -d "Body=This is a test from RestComm" -d "X-
MyCustomHeader-1=Value1" -d "X-MyCustomHeader-2=Value2"
```

Get list of SMS Messages.

This will display list of message sent

From the bash terminal, you can run the command below:

```
curl -X GET http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/SMS/Messages
```

4.8. Recordings

Recordings are generated when you use the <Record> verb. Those recordings are hosted with RestComm for you to retrieve. The Recordings list resource represents the set of an account's recordings.

Recording Resource URI. /2012-04-24/Accounts/{AccountSid}/Recordings/{RecordingSid}

To download the audio file just append .wav after the RecordingSid.

Table 4.22. Resource Properties

Property	Description
Sid	A string that uniquely identifies this recording.
DateCreated	The date that this recording was created.
DateUpdated	The date that this recording was last updated.
AccountSid	The unique id of the Account that created this recording.
CallSid	The unique id of the call during which the recording was made.
Duration	The length of the recording, in seconds.
ApiVersion	The API version in use during the recording.
Uri	The URI for this account, relative to http://localhost:port/restcomm.

4.8.1. Supported Operations

HTTP GET. Returns the representation of a Recording resource, including the properties above.

HTTP DELETE. Removes the recording from the account.

4.8.2. Recording List Resource

Recording List Resource URI. /2012-04-24/Accounts/{AccountSid}/Recordings

4.8.2.1. Supported Operations

HTTP GET. Returns the list representation of all the Recording resources for this Account, including the properties above.

> **How to Record a Message**
>
> Go to the Advanced Chapter under Section 7.1.2, “Record Verb” section to learn how to record a message.

Get List of Recordings.

The list of recorded wav files can be found in the directory $RESTCOMM_HOME/standalone/deployments/restcomm.war/recordings/

From the bash terminal, you can run the command below:

```
curl                   -X                   GET                                        http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/Recordings
```

4.9. Transcriptions

A Transcription resource represents a transcription of a recording. A transcription is a text version of a recording produced using automatic speech recognition.

Transcription Resource URI. /2012-04-24/Accounts/{AccountSid}/Transcriptions/{TranscriptionSid}

Table 4.23. Resource Properties

Property	Description
Sid	A string that uniquely identifies this transcription.
DateCreated	The date that this transcription was created.
DateUpdated	The date that this transcription was last updated.
AccountSid	The unique id of the Account that created this transcription.
Status	A string representing the status of the transcription. Possible values are in-progress, completed, and failed.
RecordingSid	The unique id of the Recording this Transcription was made of.
Duration	The duration of the transcribed audio, in seconds.
TranscriptionText	The text content of the transcription.
Uri	The URI for this account, relative to http://localhost:port/restcomm.

4.9.1. Supported Operations

HTTP GET. Returns the representation of a Transcription resource, including the properties above.

HTTP DELETE. Removes the Transcription from the account.

4.9.2. Transcription List Resource

Transcription List Resource URI. /2012-04-24/Accounts/{AccountSid}/Transcriptions

4.9.2.1. Supported Operations

HTTP GET. Returns the list representation of all the Transcription resources for this Account, including the properties above.

4.10. Notifications

A Notification resource represents a single log entry made by RestComm while handling your calls or your use of the Restful APIs. It is very useful for debugging purposes. The Notifications list resource represents the set of notifications generated for an account.

Notification Resource URI. /2012-04-24/Accounts/{AccountSid}/Notifications/{NotificationSid}

Table 4.24. Resource Properties

Property	Description
Sid	A string that uniquely identifies this transcription.
DateCreated	The date that this transcription was created.
DateUpdated	The date that this transcription was last updated.
AccountSid	The unique id of the Account that created this transcription.
CallSid	CallSid is the unique id of the call during which the notification was generated. Empty if the notification was generated by the Restful APIs without regard to a specific phone call.
ApiVersion	The RestComm API version in use when this notification was generated. May be empty for events that don't have a specific API version.
Log	An integer log level corresponding to the type of notification: 0 is ERROR, 1 is WARNING.
ErrorCode	A unique error code for the error condition. You can lookup errors, in our Error Dictionary.
MoreInfo	A URL for more information about the error condition. The URL is a page in our Error Dictionary.
MessageText	The text for the notification.

Property	Description
MessageDate	The date the notification was actually generated
RequestUrl	The URL of the resource that caused the notification to be generated.
RequestMethod	The HTTP method in use for the request that caused the notification to be generated.
RequestVariables	The HTTP GET or POST variables that RestComm generated and sent to your server. Also, if the notification was generated by the Restful APIs, this field will include any HTTP POST or PUT variables you sent.
ResponseHeaders	The HTTP headers returned by your server.
ResponseBody	The HTTP body returned by your server.
Uri	The URI for this account, relative to http://localhost:port/restcomm.

4.10.1. Supported Operations

HTTP GET. Returns the representation of a Notification resource, including the properties above.

4.10.2. Notification List Resource

Notification List Resource URI. /2012-04-24/Accounts/{AccountSid}/Notifications

4.10.2.1. Supported Operations

HTTP GET. Returns the list representation of all the Notification resources for this Account, including the properties above.

4.11. OutboundProxy

Using Outbound proxy endpoint you can get the details of Primary and Backup outbound proxy, get the current active outbound proxy and also switch outbound proxy.

OutboundProxy Resource URI. /2012-04-24//Accounts/{accountSid}/OutboundProxy

4.11.1. Supported Operations

HTTP GET. Returns the list of outbound proxies.

HTTP GET. /2012-04-24//Accounts/{accountSid}/OutboundProxy/switchProxySwitch the outbound proxy and returns the proxy in use

HTTP GET. /2012-04-24//Accounts/{accountSid}/OutboundProxy/getActiveProxyReturns the currently active outbound proxy

4.12. OutboundProxy

Using Outbound proxy endpoint you can get the details of Primary and Backup outbound proxy, get the current active outbound proxy and also switch outbound proxy.

OutboundProxy Resource URI. /2012-04-24//Accounts/{accountSid}/OutboundProxy

4.12.1. Supported Operations

HTTP GET. Returns the list of outbound proxies.

HTTP GET. /2012-04-24//Accounts/{accountSid}/OutboundProxy/switchProxySwitch the outbound proxy and returns the proxy in use

HTTP GET. /2012-04-24//Accounts/{accountSid}/OutboundProxy/getActiveProxyReturns the currently active outbound proxy

4.13. UssdPush

A UssdPush resource represents a message sent from Restcomm to a USSD gateway.

UssdPush Resource URI. /2012-04-24/Accounts/{AccountSid}/UssdPush

4.13.1. Supported Operations

HTTP POST. Sends a Ussd message to the configured USSD Gataeway

Example of UssdPush.

```
  The USSD gateway to which Restcomm must send the UssdMessage must be configured in the
restcomm.xml file. IP address and port must be configured. Username/password for the USSD are
optional.

 <!-- TelScale USSD Gateway -->
 <ussd-gateway>
 <ussd-gateway-uri>IP_USSD_GW:PORT_NUMBER</ussd-gateway-uri>
 <ussd-gateway-user></ussd-gateway-user>
 <ussd-gateway-password></ussd-gateway-password>
 </ussd-gateway>



 See below a curl example for the USSD Push:
        curl -X POST -H "application/json" http://
ACae6e420f425248d6a26948c17a9e2acf:YOUR_PWD@RESTCOMM_IP_ADDRESS:8080/restcomm/2012-04-24/
Accounts/ACae6e420f425248d6a26948c17a-d "From=Restcomm" -d "To=123" -d "Url=http://
RESTCOMM_IP_ADDRESS:8080/restcomm-rvd/services/apps/YOUR_USSD_APP/controller"
```

Chapter 5. Restcomm Admin User Interface

The admin user interface is a great way to perform repetive Restcomm task in a orderly manner. Care has been taken to make the user experience intuitive for those new to the platform. In this chapter, you will learn how to use some of the features available to create Clients, Phone Numbers, Check Call Logs, Get Speech-to-Text Transcription and many more

Please also visit the official Telestax documentation page for more usage tutorials:http://www.telestax.com/docs

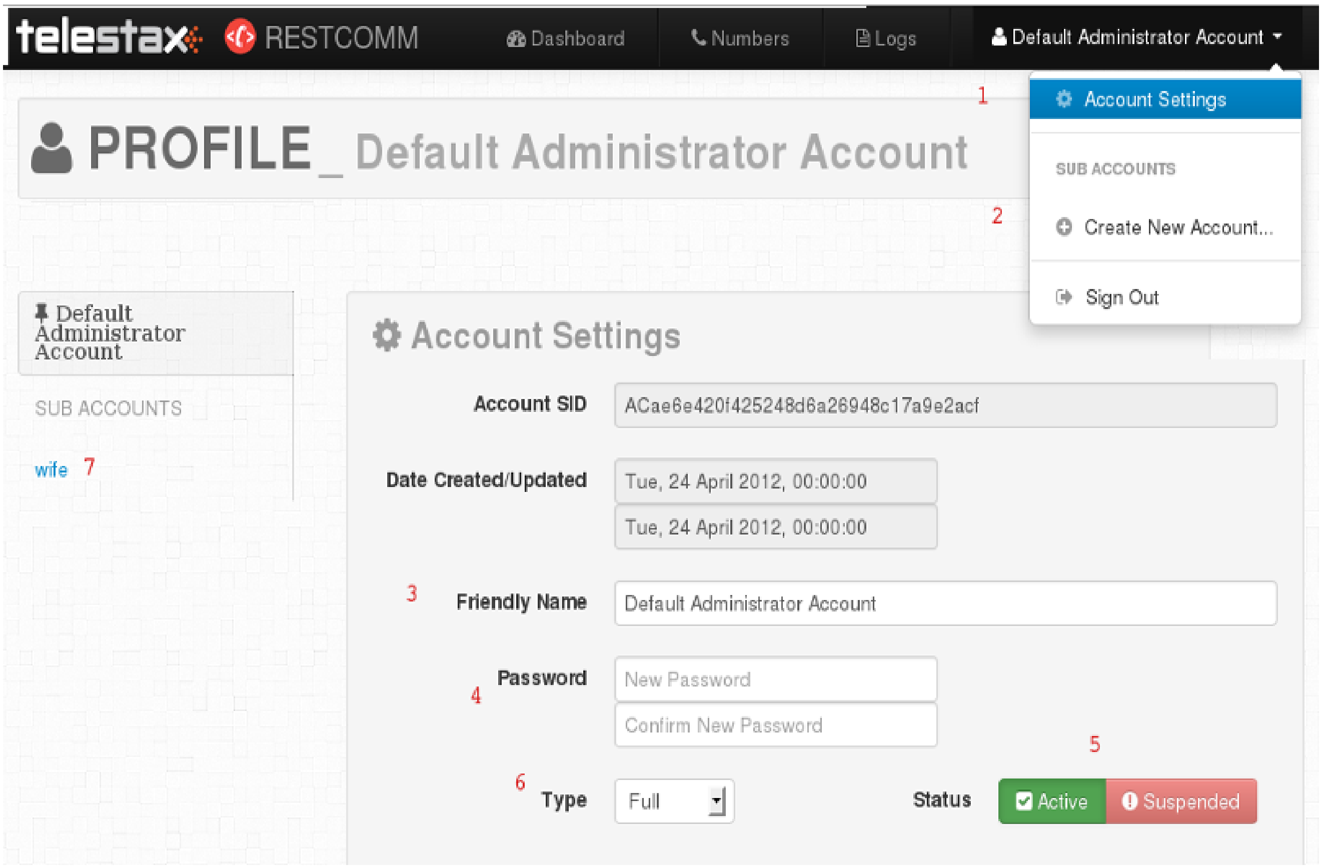
5.1. Login Interface

You need to make sure Restcomm is running before you can use the Admin User Interface. When you are running on a local install, open your web browser and go to this url http://127.0.0.1:8080/restcomm-management [http://127.0.0.1:8080/restcomm-management]. You will see a screenshot similar to the one below

Please see the Section 2.2, “Login into Restcomm and Changing the Default Password” for more information about changing the default password.

5.2. Passwords, Sub Accounts Settings

When you click on the Administrators Account at the top right corner of the window, as shown in the screenshot below, you will be able to change your Administrators passwords and create Sub Accounts.



Account SID: ACae6e420f425248d6a26948c17a9e2acf

Date Created/Updated: Tue, 24 April 2012, 00:00:00 / Tue, 24 April 2012, 00:00:00

Friendly Name: Default Administrator Account

Password: New Password / Confirm New Password

Type: Full

Status: Active | Suspended

Table 5.1. Account Settings

Number	Description
1	Shows the current Acount profile
2	You can create new Sub Accounts
3	Descriptive name of the user account
4	New password of the current logged in account
5	You can activate or suspend an account
6	There are three options, Trial, Basic and Full(default)
7	List of Sub Accounts

5.3. Dashboard

This is the default page where you can get an overview of your Restcomm Installation.

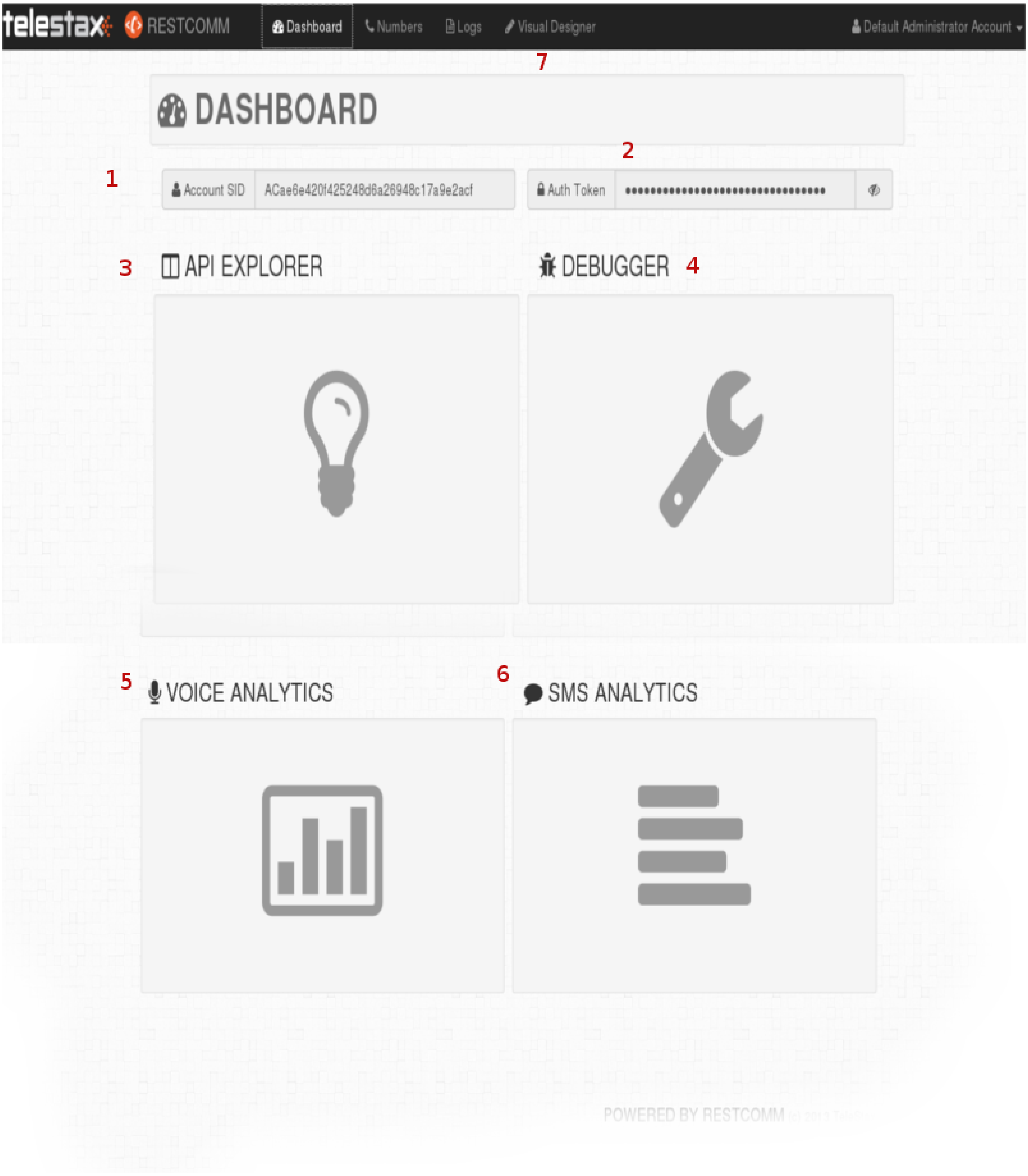


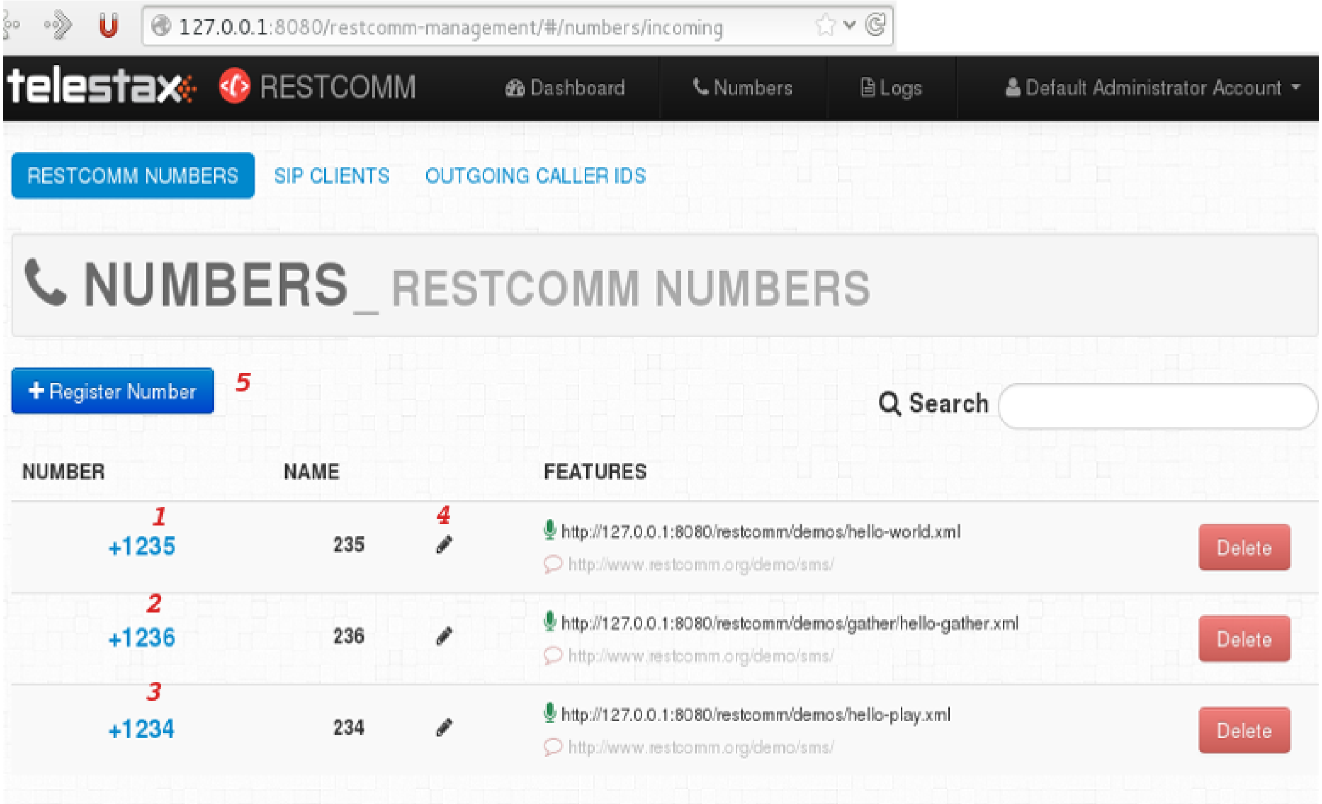
Table 5.2. Dashboard

Number	Description
1	Account SID is the default used by the Administrator's account

Number	Description
2	The Auth Token is the password that is required for any Restcomm operation. You can click on the hidden button to reveal the hashed password.
3	The Debugger lets you troubleshoot any issues you might encounter using Restcomm
4	A quick way to get all the API exposed by Restcomm
5	Additional information about Calls parsed by Restcomm
6	Additional information about SMS parsed by Restcomm
7	This will take you to the Restcomm Visual Designer (RVD) page located a http://127.0.0.1:8080/restcomm-rvd

5.4. Restcomm Numbers

This will show the default demo applications that come with Restcomm. When you start creating applications and attaching numbers, they will also be displayed on this page.



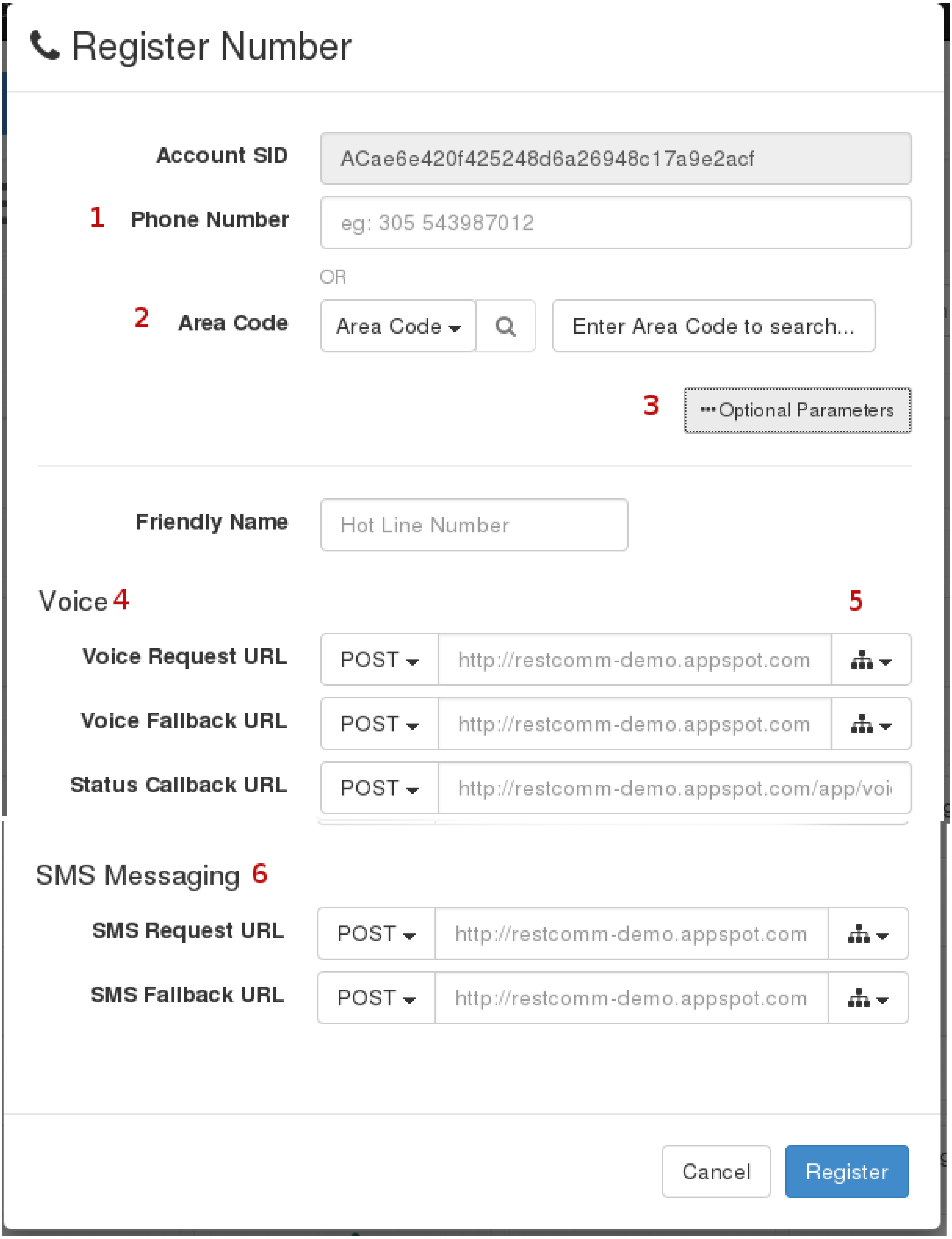
NUMBER	NAME	FEATURES
+1235	235	http://127.0.0.1:8080/restcomm/demos/hello-world.xml http://www.restcomm.org/demo/sms/
+1236	236	http://127.0.0.1:8080/restcomm/demos/gather/hello-gather.xml http://www.restcomm.org/demo/sms/
+1234	234	http://127.0.0.1:8080/restcomm/demos/hello-play.xml http://www.restcomm.org/demo/sms/

Table 5.3. Restcomm Numbers

Number	Description
1	The number 1235 is attached to the hello-world.xml application. You must have configured VoiceRSS text-to-speech to use this application
2	The number 1236 is attached to a the Gather verb. It will ask you to enter a number and you can hear the number you enter. You must have configured VoiceRSS text-to-speech to use this application
3	The number 1234 plays a pre-recorded file.
4	This icon lets you edit the Name of the entry to a more descriptive one.
5	This button lets you create a new number that can be attached to a RCML

5.4.1. Register Number

When you click on the Register Number button, you will see a screenshot similar to the one below. This will allow you to create a new phone number that can be attached to a Restcomm application.



☎ Register Number

Account SID: ACae6e420f425248d6a26948c17a9e2acf

Phone Number: eg: 305 543987012

OR

Area Code: Area Code ▾ | Enter Area Code to search...

Friendly Name: Hot Line Number

Voice

Voice Request URL: POST ▾ http://restcomm-demo.appspot.com

Voice Fallback URL: POST ▾ http://restcomm-demo.appspot.com

Status Callback URL: POST ▾ http://restcomm-demo.appspot.com/app/voi

SMS Messaging

SMS Request URL: POST ▾ http://restcomm-demo.appspot.com

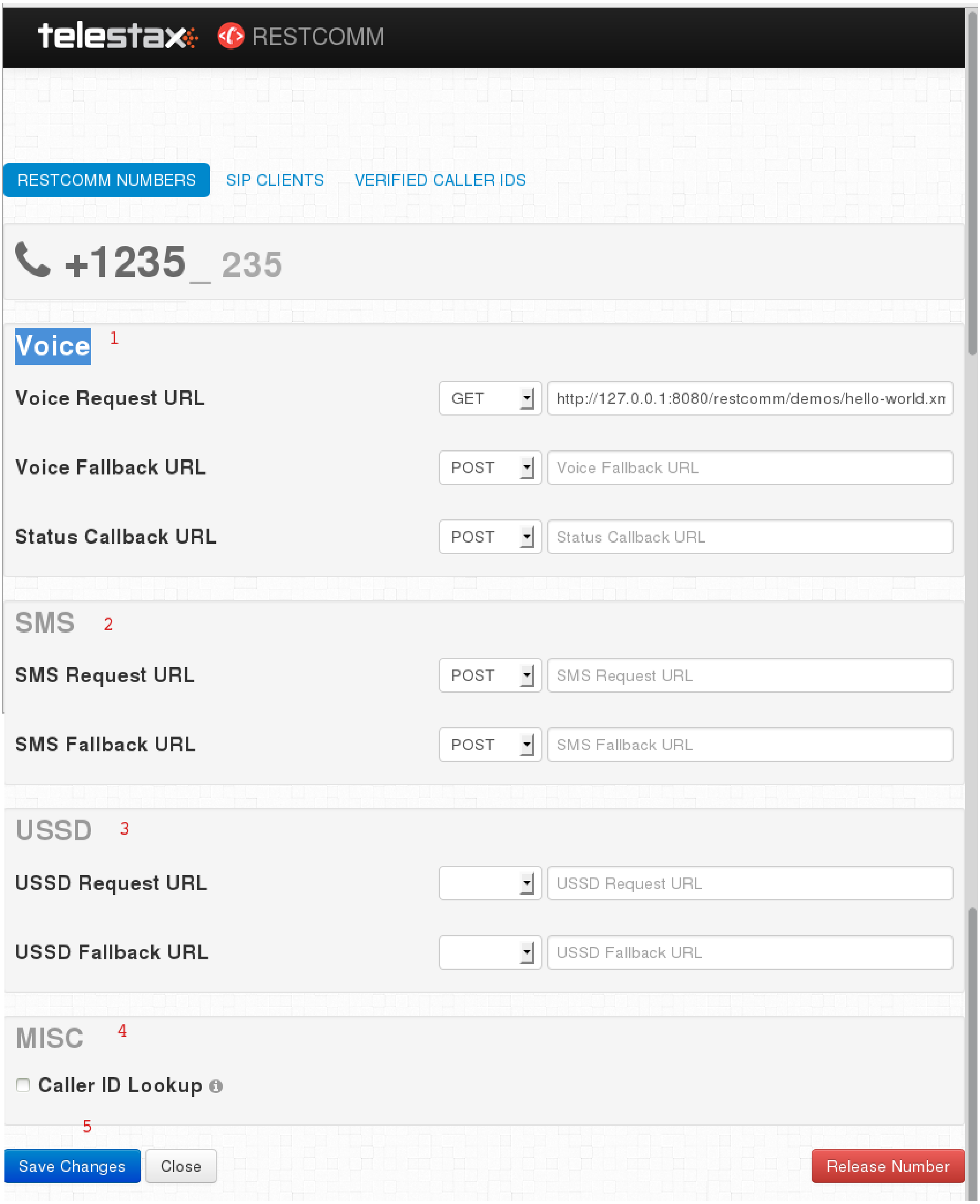
SMS Fallback URL: POST ▾ http://restcomm-demo.appspot.com

Table 5.4. Restcomm Numbers

Number	Description
1	The field to enter the phone number.
2	This button will show the Area Code. You must have Voip Innovations API account configured in the restcomm.xml file to use this feature. It will search for DID in the area code specified. At the moment, only US DIDs are available.
3	This button will show advanced options if you want to add more features to the phone number like the VoiceUrl
4	See the Rest API Chapter 4, *Restful APIs* for more information
5	This button will show available Restcomm Visual Designer (RVD) applications that can be linked to a phone number.
6	See the Rest API Chapter 4, *Restful APIs* for more information

5.4.2. Edit Update Number

Editing a phone number can be done by clicking on the number, the screenshot below shows how you can edit the number 1235. You can change the VoiceUrl to which the number is attached.



☎ +1235_ 235

Voice

Voice Request URL: GET http://127.0.0.1:8080/restcomm/demos/hello-world.xn

Voice Fallback URL: POST Voice Fallback URL

Status Callback URL: POST Status Callback URL

SMS

SMS Request URL: POST SMS Request URL

SMS Fallback URL: POST SMS Fallback URL

USSD

USSD Request URL: USSD Request URL

USSD Fallback URL: USSD Fallback URL

MISC

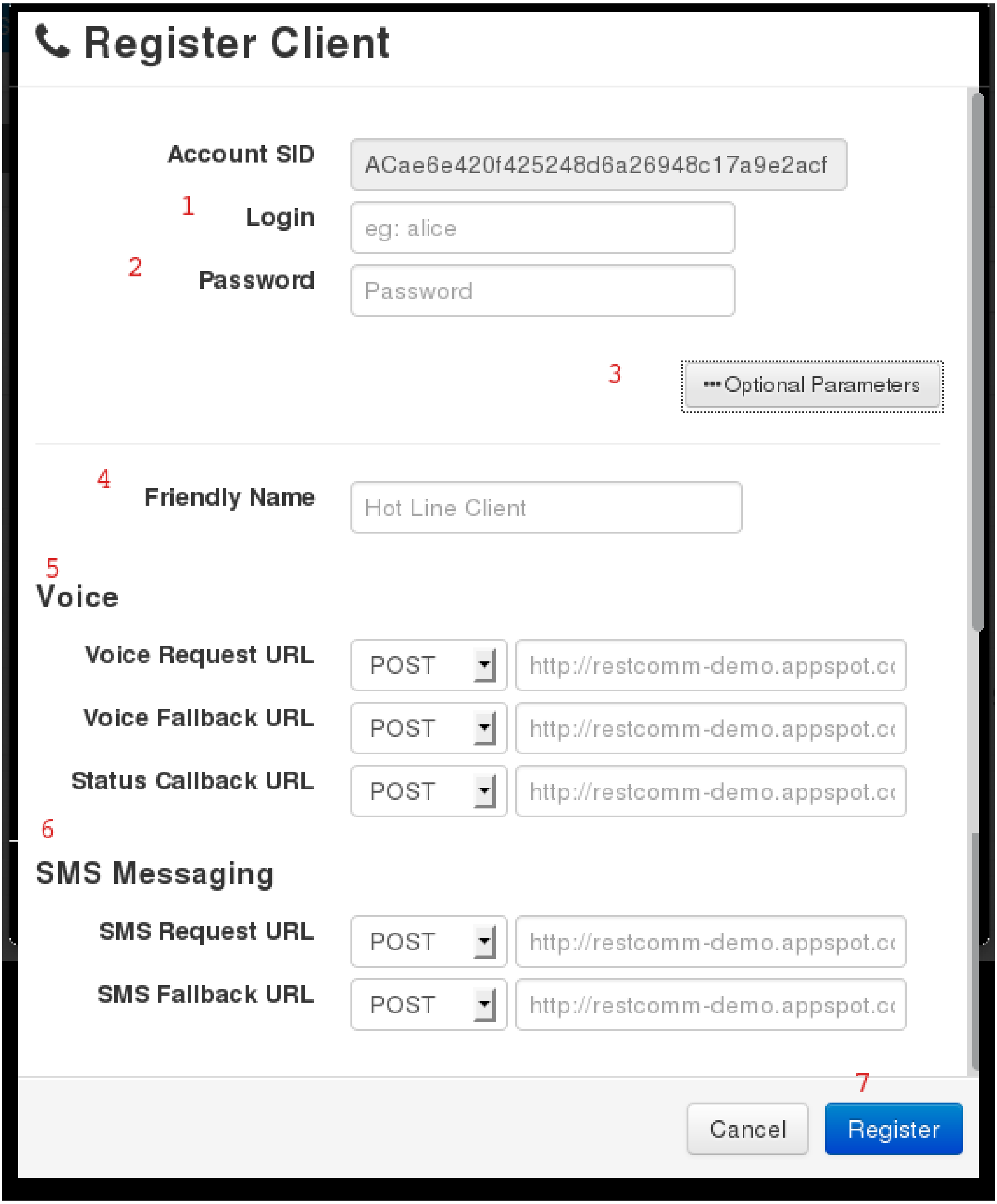
☐ Caller ID Lookup ℹ

Table 5.5. Edit Update Numbers

Number	Description
1	You can link the phone number to a VoiceUrl application. See the REST API for more details.
2	You can link the phone number to a SMS application. See the REST API for more details
3	You can link the phone number to a USSD application. See the REST API for more details
4	Caller Id lookup requires a CNAM provier
5	You can save your changes or press close to discard the changes

5.5. SIP Clients

SIP Client is a feature that allows you to create a Restcomm profile that can be used for P2P or B2BUA calls. This will be empty until you create a new client. You can create a new client by clicking on the Resgister SIP Clent button.



☎ Register Client

Account SID: ACae6e420f425248d6a26948c17a9e2acf

Login: eg: alice

Password: Password

Friendly Name: Hot Line Client

Voice

Voice Request URL: POST http://restcomm-demo.appspot.c

Voice Fallback URL: POST http://restcomm-demo.appspot.c

Status Callback URL: POST http://restcomm-demo.appspot.c

SMS Messaging

SMS Request URL: POST http://restcomm-demo.appspot.c

SMS Fallback URL: POST http://restcomm-demo.appspot.c

Table 5.6. SIP Clients

Number	Description
1	The client name. (ex. alice or bob)
2	The password that will be used to when you want to register the client with restcomm
3	Use to open optional parameters windows
4	This can be the full name of the client (ex.Alice Wilkinson) or any descriptive name
5	This is where you specify the VoiceUrl that is automatically invoked when the client is called. See the REST API for more details.
6	See the REST API for more details.
7	This will validate your changes and create the client.

5.6. Outgoing CallerId

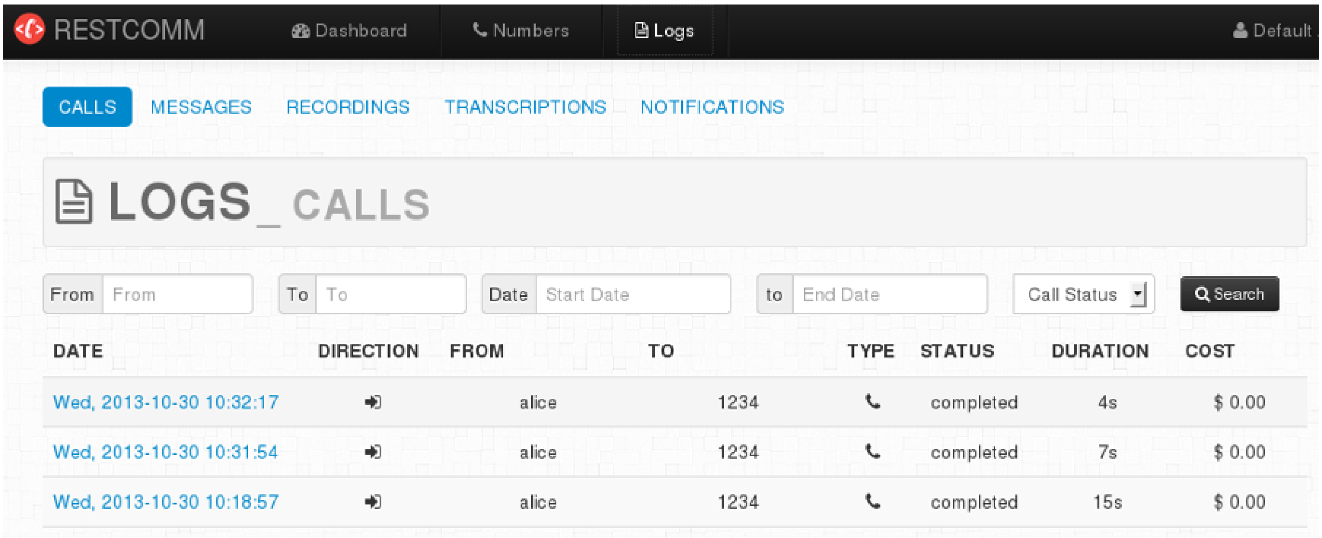
Will be available in future release

5.7. Logs

The log section gives you an overview of current Restcomm system information.

5.7.1. Logs - Calls

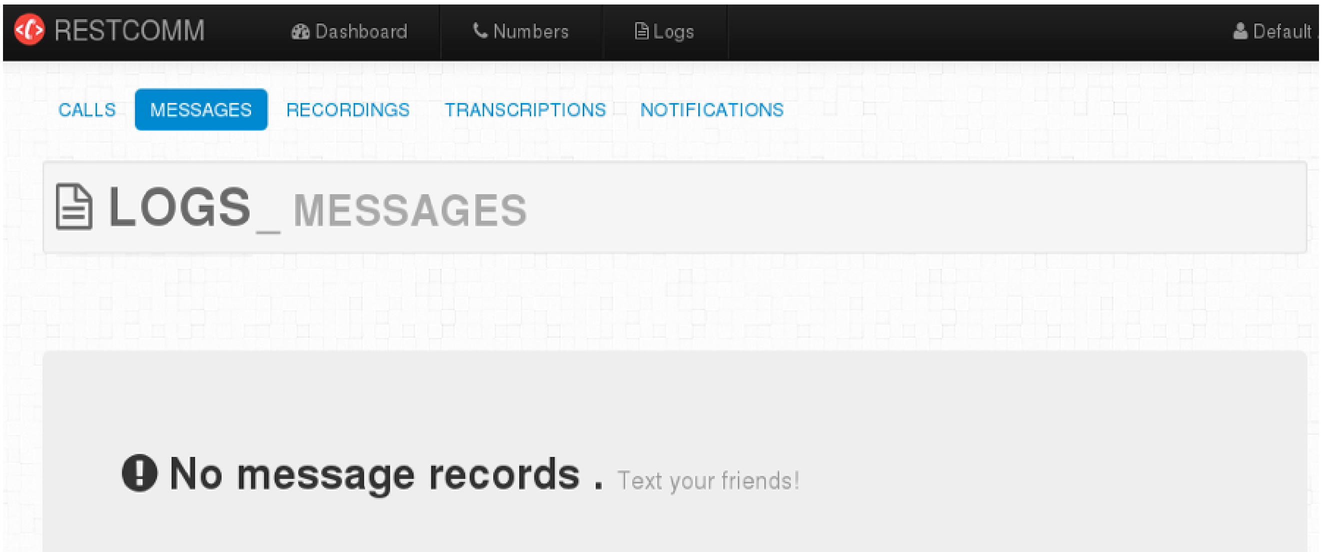
A list of all calls that have been processed by Restcomm



DATE	DIRECTION	FROM	TO	TYPE	STATUS	DURATION	COST
Wed, 2013-10-30 10:32:17	➜	alice	1234	☎	completed	4s	$ 0.00
Wed, 2013-10-30 10:31:54	➜	alice	1234	☎	completed	7s	$ 0.00
Wed, 2013-10-30 10:18:57	➜	alice	1234	☎	completed	15s	$ 0.00

5.7.2. Logs - Messages

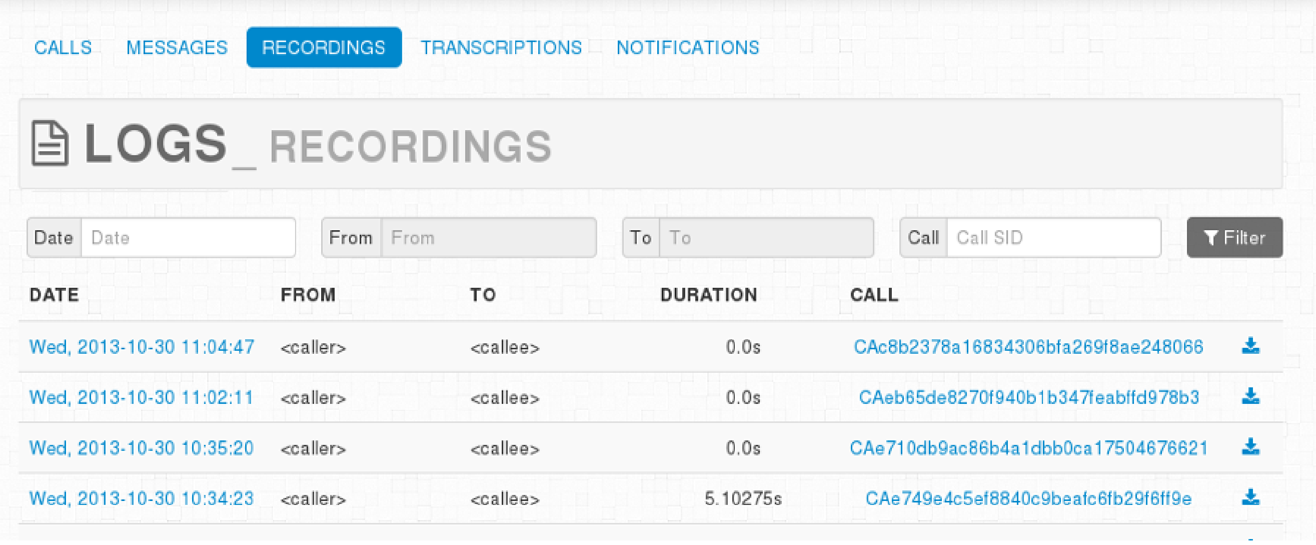
A list of all SMS messages that have been processed by Restcomm



No message records . Text your friends!

5.7.3. Logs - Recordings

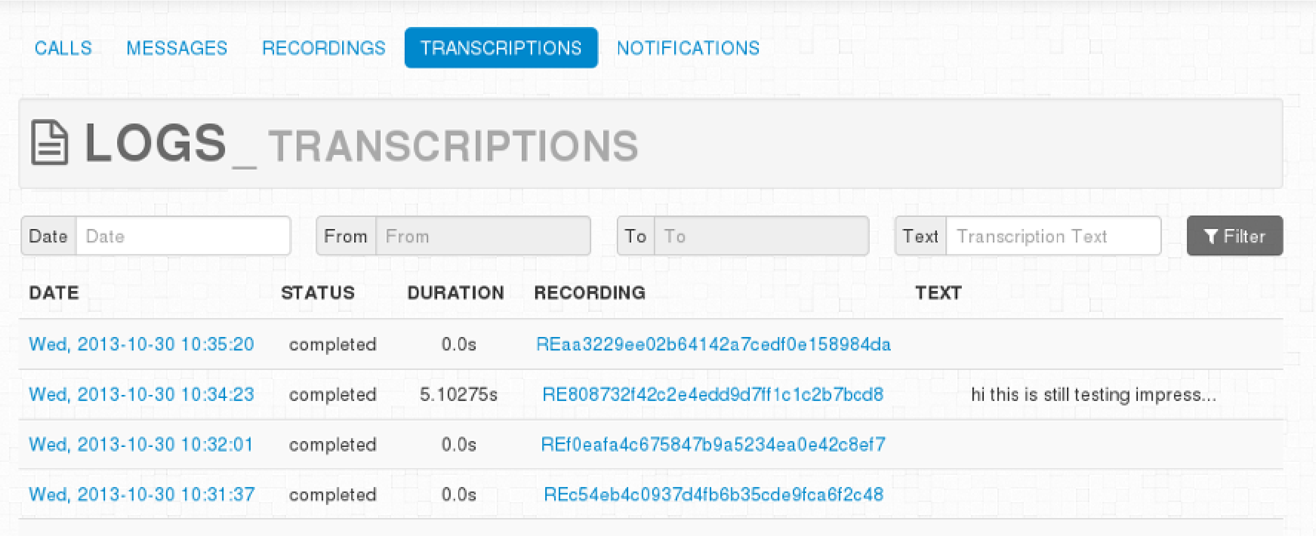
A list of all Recordings (using the Record Verb) that have been processed by Restcomm



DATE	FROM	TO	DURATION	CALL
Wed, 2013-10-30 11:04:47	<caller>	<callee>	0.0s	CAc8b2378a16834306bfa269f8ae248066
Wed, 2013-10-30 11:02:11	<caller>	<callee>	0.0s	CAeb65de8270f940b1b347feabffd978b3
Wed, 2013-10-30 10:35:20	<caller>	<callee>	0.0s	CAe710db9ac86b4a1dbb0ca17504676621
Wed, 2013-10-30 10:34:23	<caller>	<callee>	5.10275s	CAe749e4c5ef8840c9beafc6fb29f6ff9e

5.7.4. Logs - Transcriptions

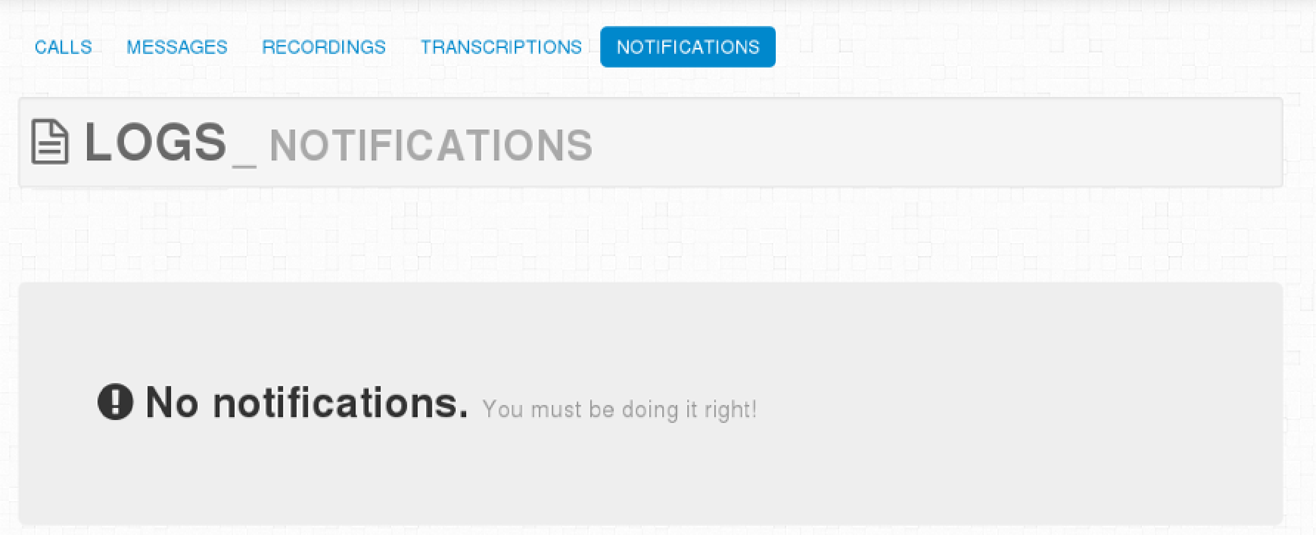
A list of all Transcriptions that have (using the Transcribe parameters of the Record Verb) that have been processed by Restcomm. See the Chapter 4, *Restful APIs*



DATE	STATUS	DURATION	RECORDING	TEXT
Wed, 2013-10-30 10:35:20	completed	0.0s	REaa3229ee02b64142a7cedf0e158984da	
Wed, 2013-10-30 10:34:23	completed	5.10275s	RE808732f42c2e4edd9d7ff1c1c2b7bcd8	hi this is still testing impress...
Wed, 2013-10-30 10:32:01	completed	0.0s	REf0eafa4c675847b9a5234ea0e42c8ef7	
Wed, 2013-10-30 10:31:37	completed	0.0s	REc54eb4c0937d4fb6b35cde9fca6f2c48	

5.7.5. Logs - Notifications

A list of all Notifications received by Restcomm.



No notifications. You must be doing it right!

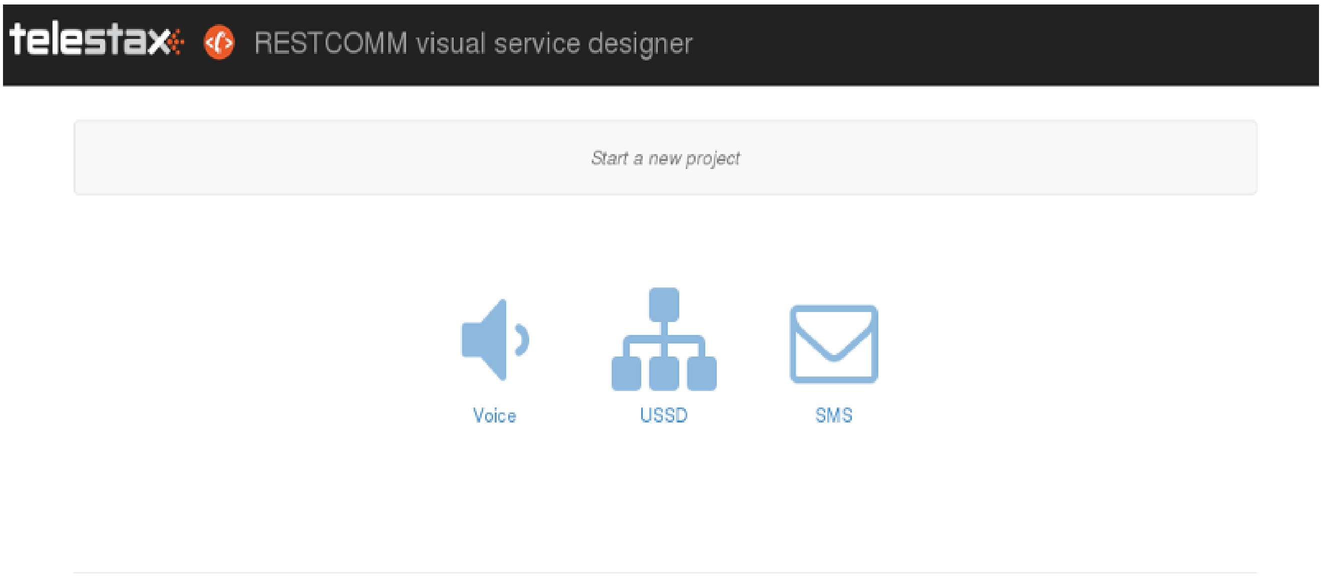
Chapter 6. Restcomm Visual Designer

Restcomm Visual Designer is a practical feature that lets you create applications easily using a GUI based interface.

Please also visit the official Telestax documentation page for more usage tutorials:http://www.telestax.com/docs

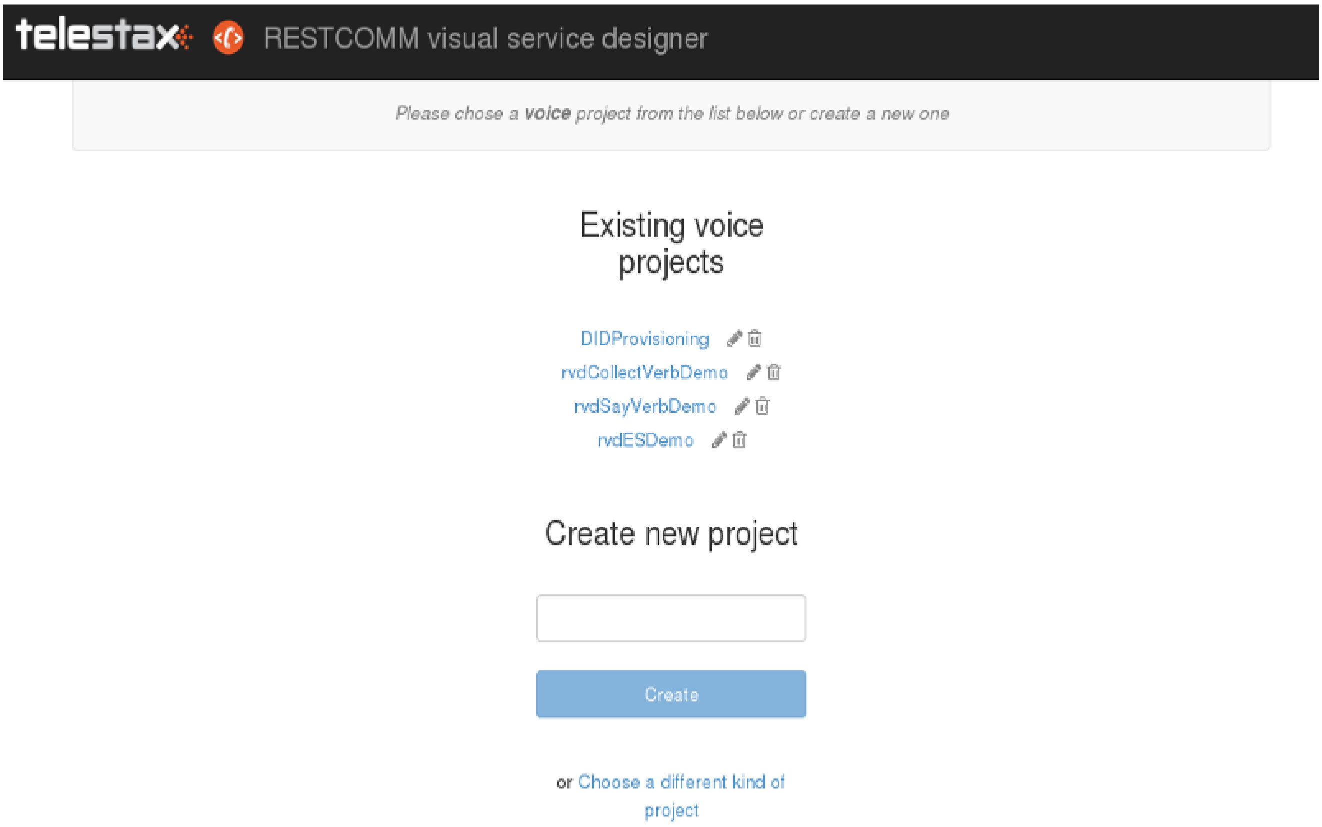
6.1. Getting Started

In order to access the designer, open your web browser and go to this url http://127.0.0.1:8080/restcomm-rvd. This page lets you select which type of project you will like to create. There are three options, Voice, USSD and SMS.



6.2. Voice Projects

In the screenshot below, some demo voice projects and you are also able to create a new project.



Please chose a **voice** *project from the list below or create a new one*

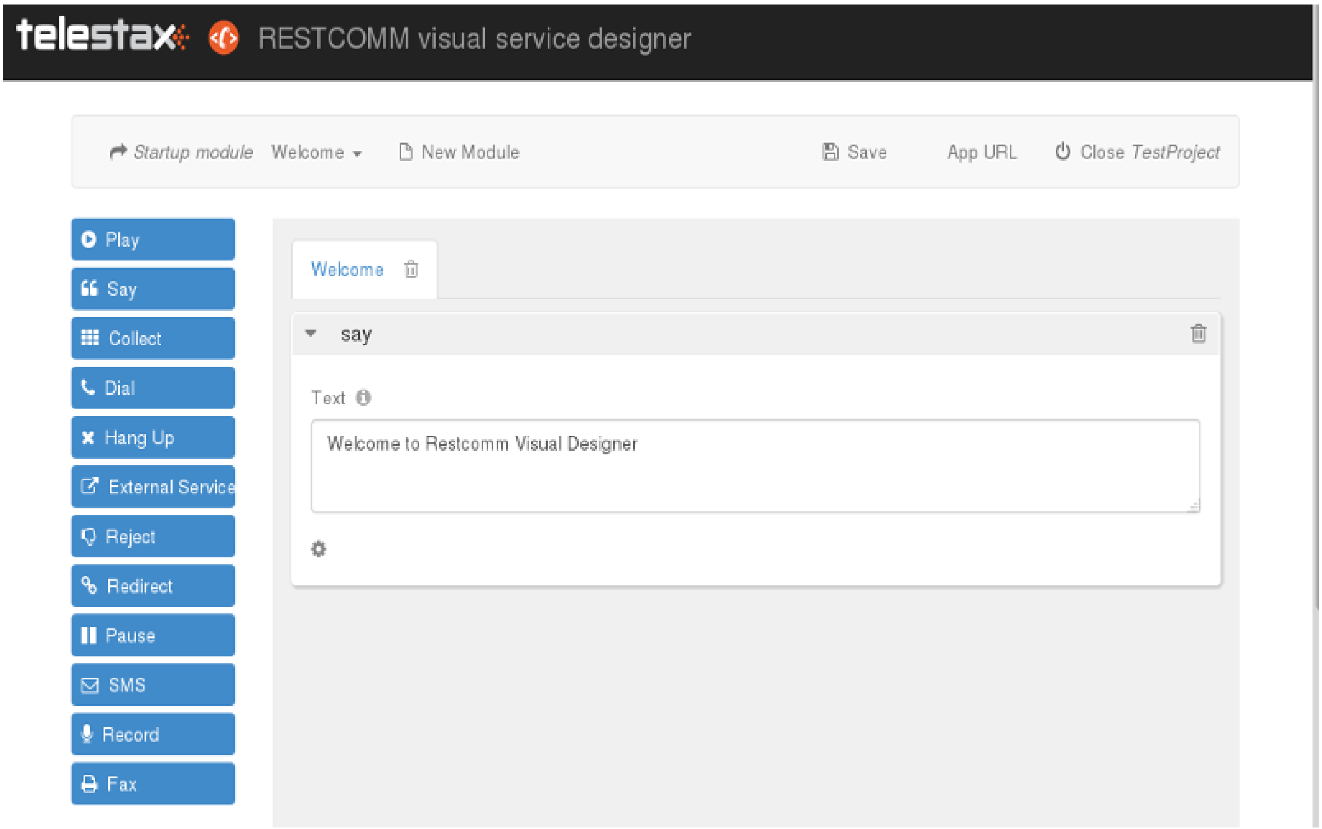
Existing voice projects

DIDProvisioning
rvdCollectVerbDemo
rvdSayVerbDemo
rvdESDemo

Create new project

or Choose a different kind of project

In the screenshot below, you can see the main dashboard of the RVD voice project. It has a drag and drop feature.

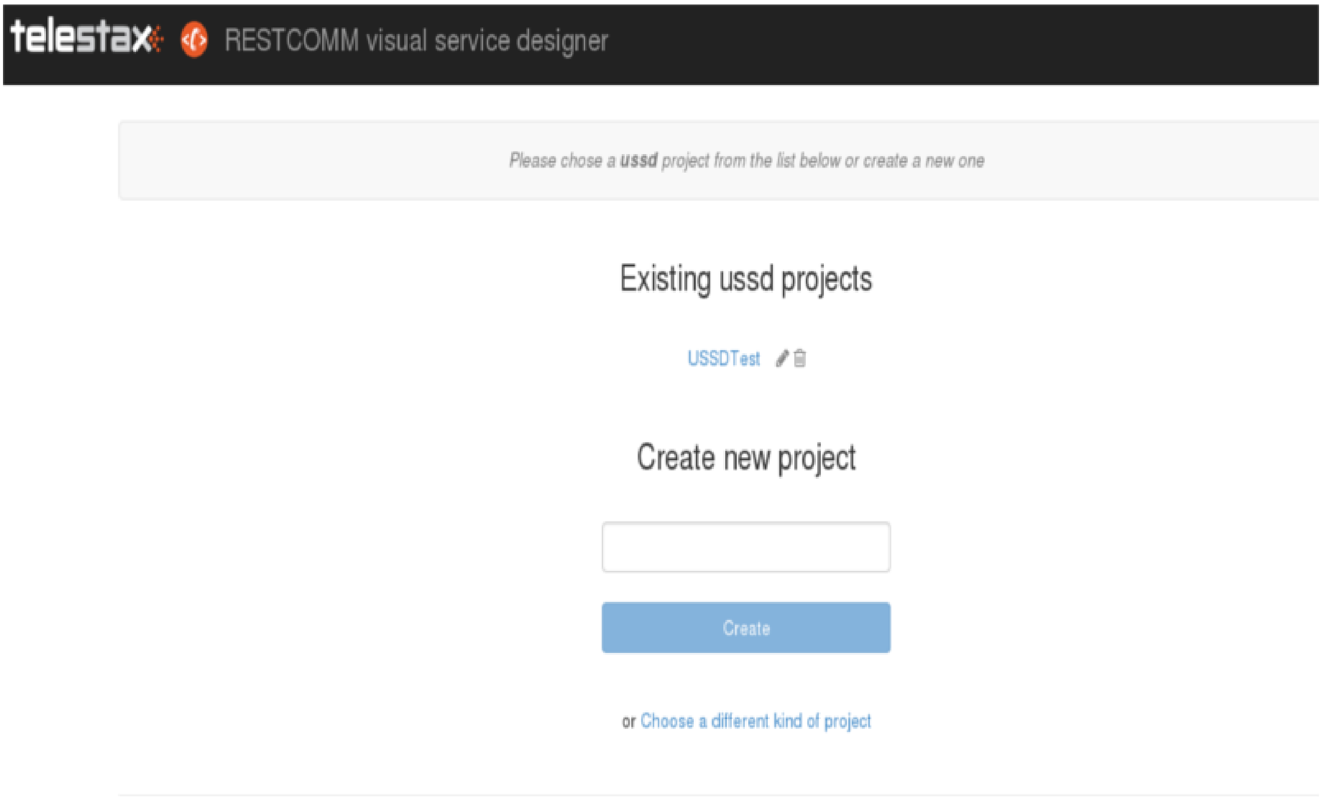


Text: Welcome to Restcomm Visual Designer

Please also visit the official Telestax documentation page for more RVD usage tutorials:http://www.telestax.com/docs

6.3. USSD Projects

The main page of the USSD Restcomm integration is similar to the screenshot below. You can create new projects or access existing projects from this window.



telestax RESTCOMM visual service designer

Please chose a **ussd** project from the list below or create a new one

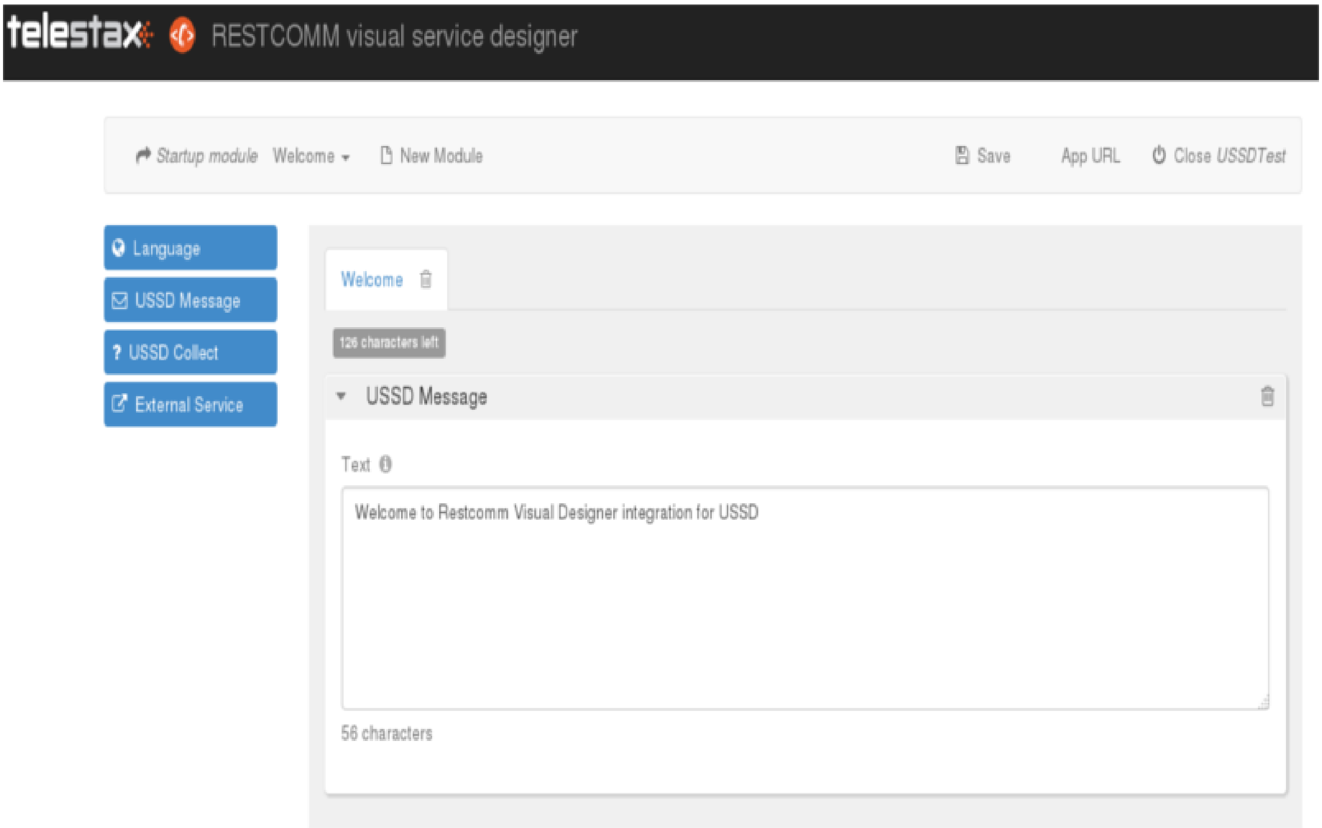
Existing ussd projects

USSDTest

Create new project

or Choose a different kind of project

In the screenshot below, you can see the main dashboard of the USSD project. It has a drag and drop feature.



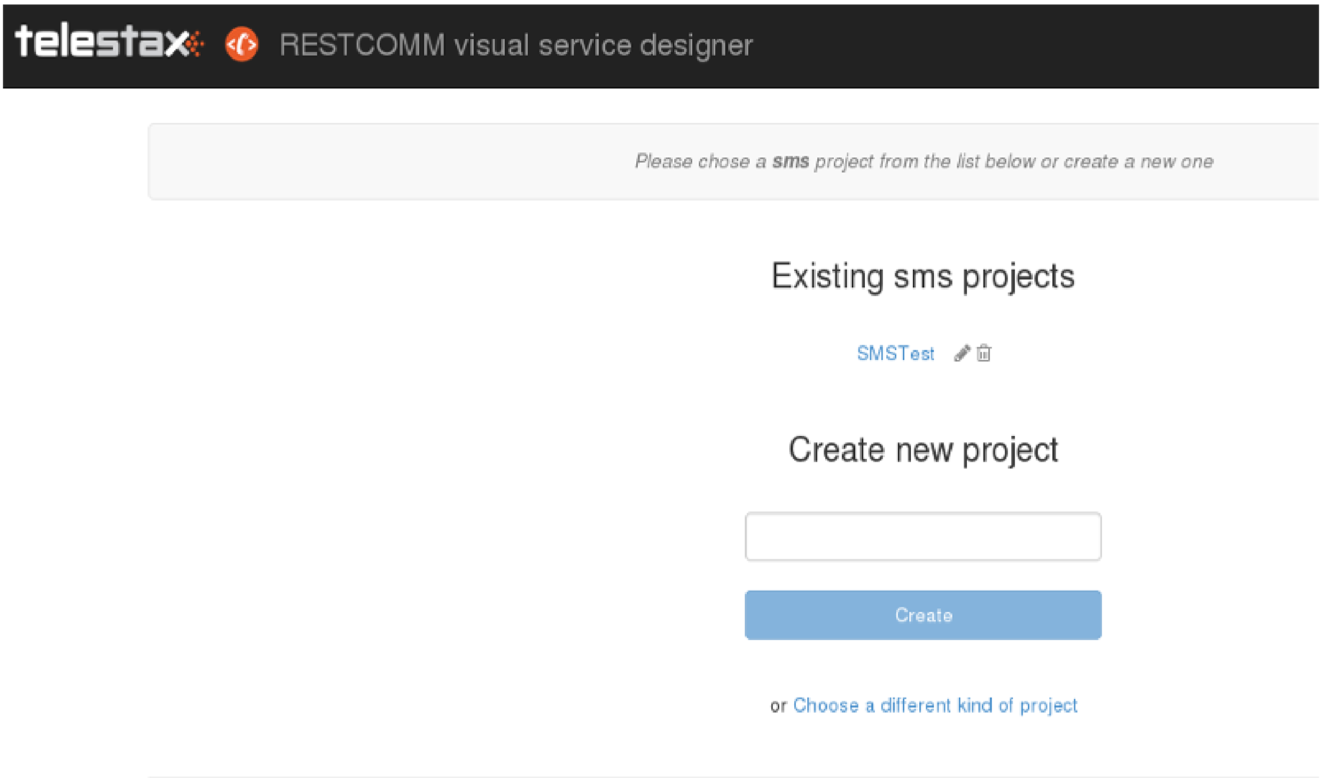
Welcome to Restcomm Visual Designer integration for USSD

56 characters

Please also visit the official Telestax documentation page for more RVD USSD usage tutorials:http://www.telestax.com/docs

6.4. SMS Projects

The main page of the SMS Restcomm integration is similar to the screenshot below. You can create new projects or access existing projects from this window.



telestax RESTCOMM visual service designer

*Please chose a **sms** project from the list below or create a new one*

Existing sms projects

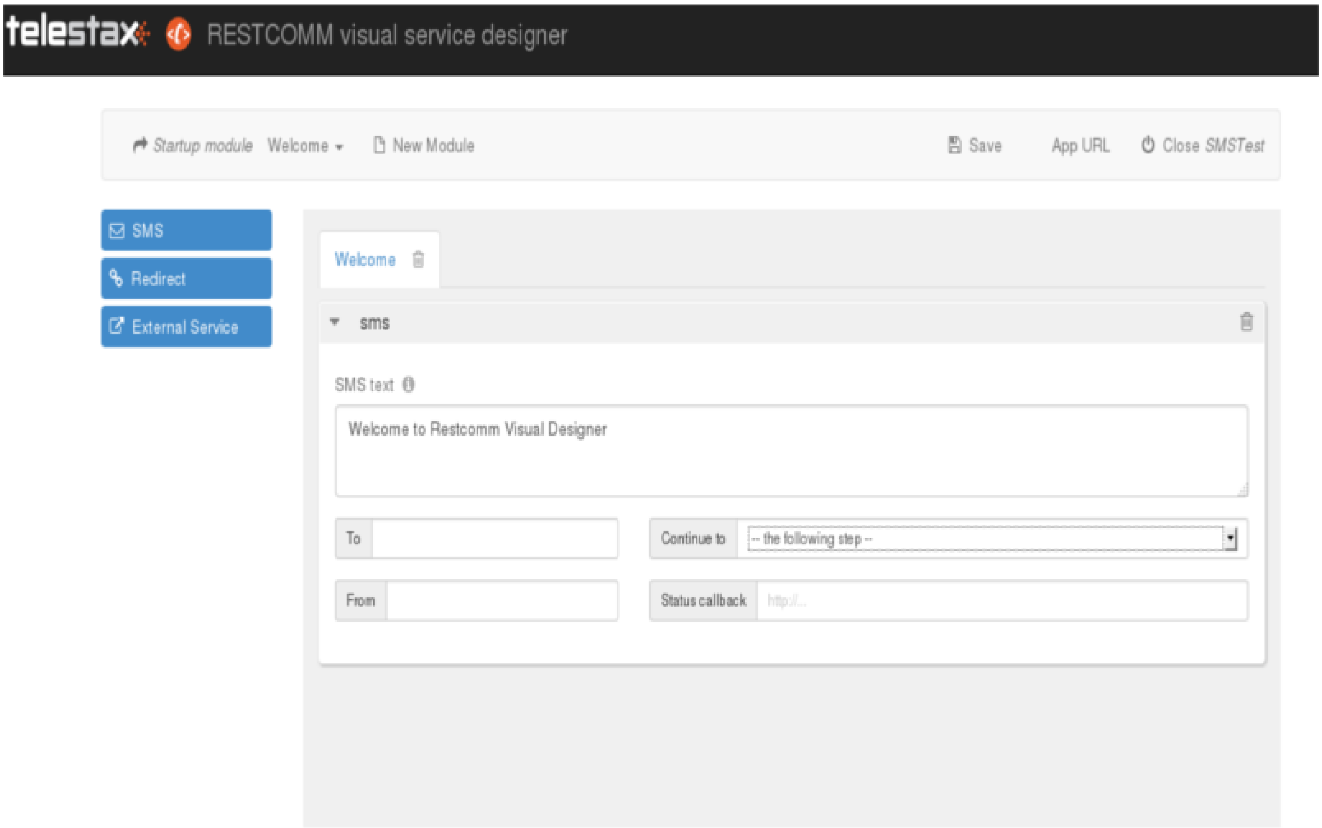
SMSTest

Create new project

Create

or Choose a different kind of project

In the screenshot below, you can see the main dashboard of the SMS project. It has a drag and drop feature.



telestax RESTCOMM visual service designer

Welcome to Restcomm Visual Designer

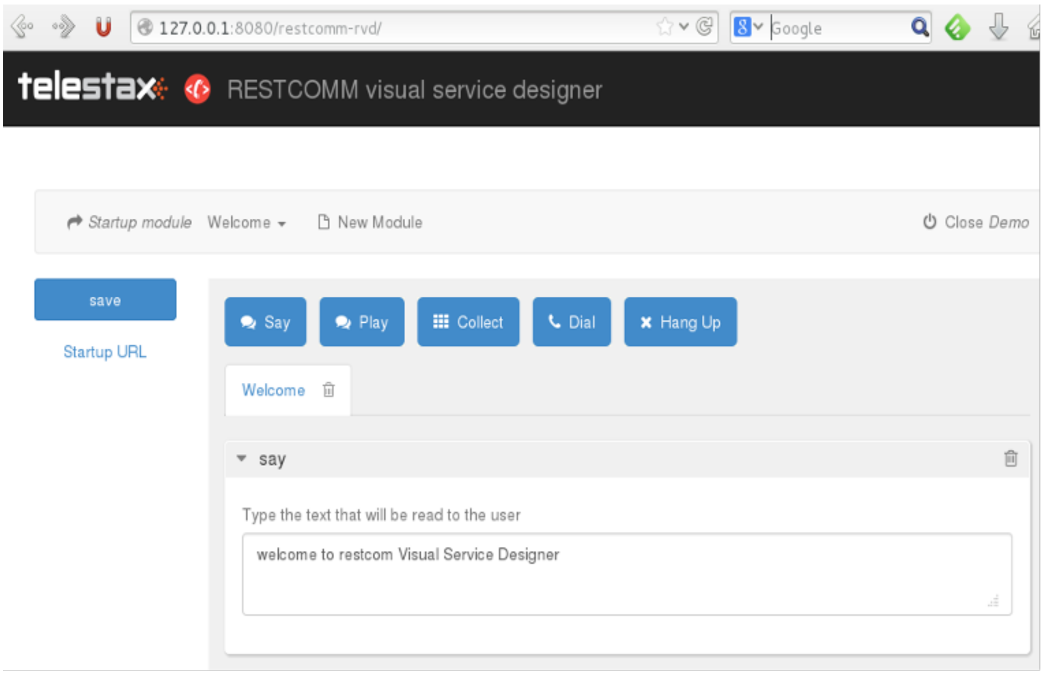
-- the following step --

Please also visit the official Telestax documentation page for more RVD SMS usage tutorials:http://www.telestax.com/docs

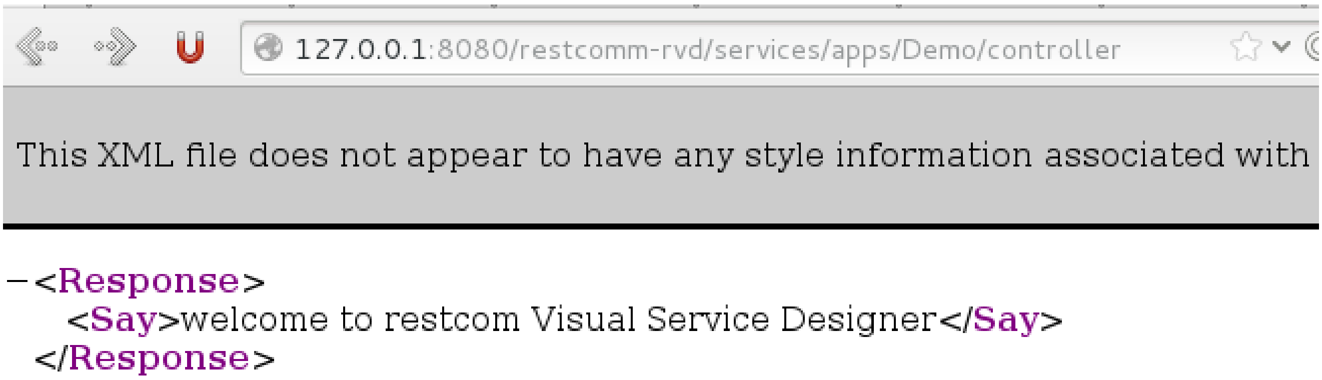
6.5. Demo Application

In this section, you will learn how to build a sample application.

Create a new project called Demo. The result will be similar to the screenshot below.


127.0.0.1:8080/restcomm-rvd/
Google
telestax
RESTCOMM visual service designer
Startup module
Welcome
New Module
Close Demo
save
Startup URL
Say
Play
Collect
Dial
Hang Up
Welcome
say
Type the text that will be read to the user
welcome to restcom Visual Service Designer

Press the save button and click on the Startup Url to make sure the project is correctly saved. The result will be similar to the screenshot below.



This XML file does not appear to have any style information associated with

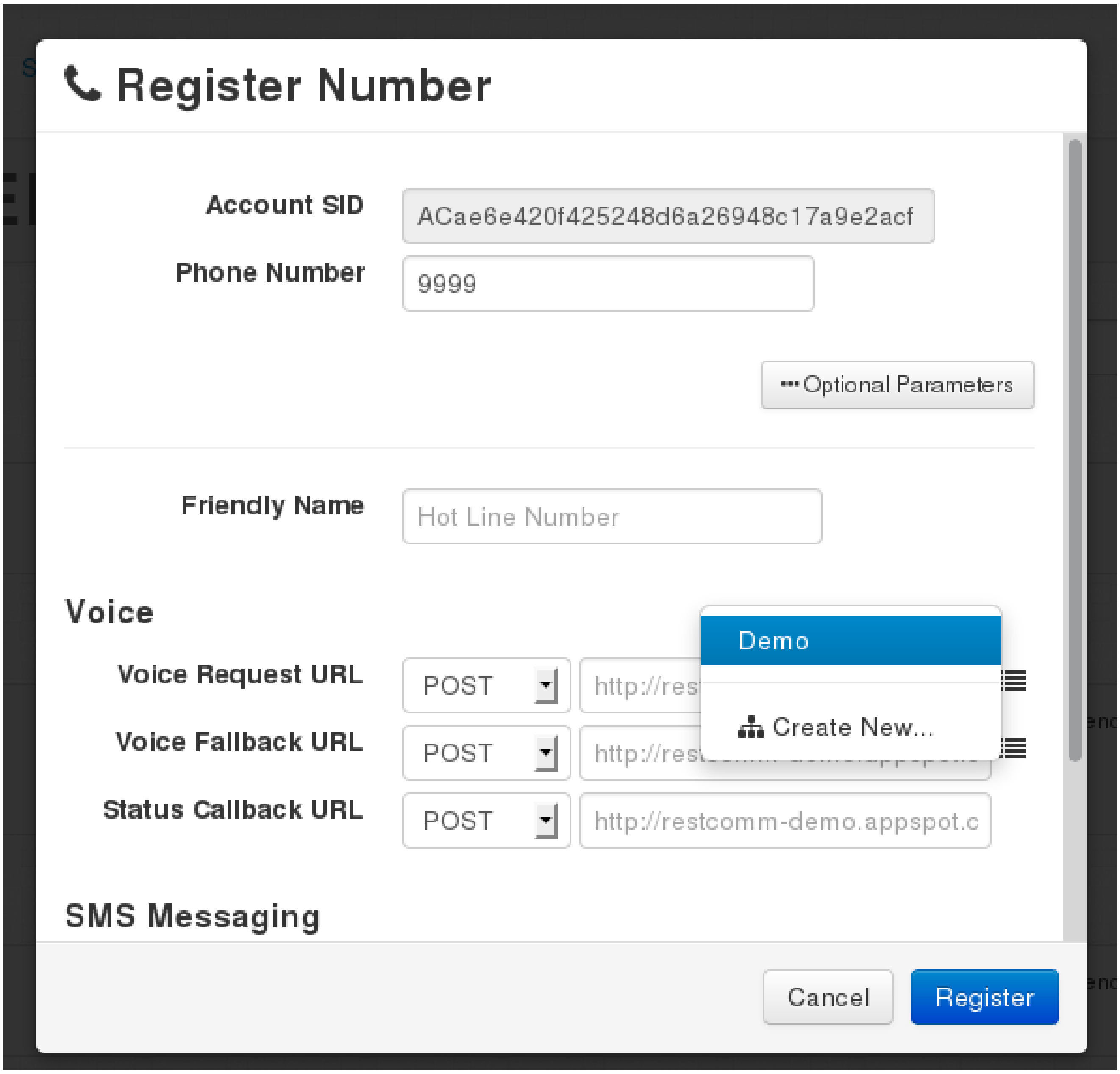
```
-<Response>
   <Say>welcome to restcom Visual Service Designer</Say>
 </Response>
```

Next you need to go to the Restcomm Admin User Interface http://127.0.0.1:8080/restcomm-management/ . Login and press the menu "Numbers", then press the button, "Register Number". You will see a window similar to the screenshot below.

Leave Account SID as default. Enter a "Phone Number" you want to use. In the demo we shall be using 9999.

Under "Voice" -> Voice Request URL, click on the List icon to reveal the Demo application you created using the Restcomm Visual Designer. See screenshot below.

Press "Register" to save your work and return to the list of available phone numbers.



☎ Register Number

Account SID: ACae6e420f425248d6a26948c17a9e2acf

Phone Number: 9999

Friendly Name: Hot Line Number

Voice

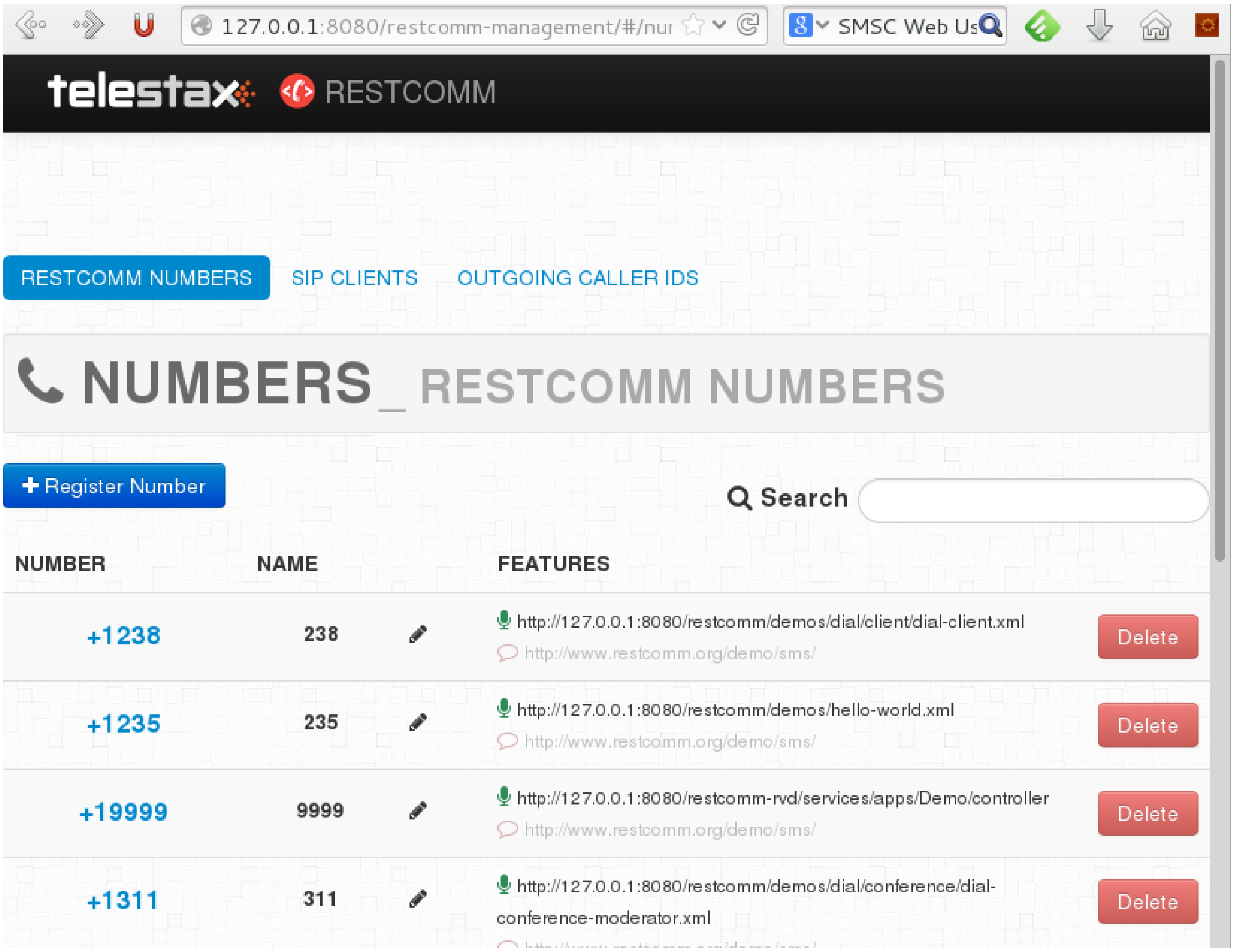
Voice Request URL: POST http://res

Voice Fallback URL: POST http://res

Status Callback URL: POST http://restcomm-demo.appspot.c

SMS Messaging

You have just created a new phone number "9999" and have linked it to your Demo application. See below.



NUMBERS_ RESTCOMM NUMBERS

NUMBER	NAME	FEATURES
+1238	238	http://127.0.0.1:8080/restcomm/demos/dial/client/dial-client.xml http://www.restcomm.org/demo/sms/
+1235	235	http://127.0.0.1:8080/restcomm/demos/hello-world.xml http://www.restcomm.org/demo/sms/
+19999	9999	http://127.0.0.1:8080/restcomm-rvd/services/apps/Demo/controller http://www.restcomm.org/demo/sms/
+1311	311	http://127.0.0.1:8080/restcomm/demos/dial/conference/dial-conference-moderator.xml

You can then start a SIP phone to test your Demo application. Note that you must have enabled Text-to-speech for this application to work. See the Getting started chapter to learn how to configure VoiceRSS.

Chapter 7. Advanced RestComm Examples

In this chapter, you will learn how to use more advanced features of Restcomm verbs to create communications applications

7.1. Activating Text-to-Speech (TTS)

You must get an API key from http://www.voicerss.org [http://www.voicerss.org/] in order to proceed with this section. You can register for a free account and an API key will be emailed to you. Once you have the API key, open the `$RESTCOMM_HOME/standalone/deployments/restcomm.war/WEB-INF/conf/restcomm.xml` file and find the speech-synthesizer VoiceRSS section. Add your API key as shown below and restart RestComm

```
<speech-synthesizer
  class="org.mobicents.servlet.restcomm.tts.VoiceRSSSpeechSynthesizer">
  <service-root>http://api.voicerss.org</service-root>
  <apikey>2901c0aXXXXXXXXXXXXXXX</apikey>
```

Once you have done that, you are ready to use Text-to-Speech feature of RestComm.

7.1.1. Say Verb

Create a file called test.xml in the `$RESTCOMM_HOME/standalone/deployments/restcomm.war/demos/` directory and copy the content of the application below into the `test.xml` file and save it

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
        <Say>Welcome to RestComm, you have successfully tested the Say Verb</Say>
</Response>
```

From a command terminal run the Curl command below to bind the 5555 phone number to the test.xml file.

```
curl                   -X                   POST                                      http://
ACae6e420f425248d6a26948c17a9e2acf:77f8c12cc7b8f8423e5c38b035249166@127.0.0.1:8080/
```

```
restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/IncomingPhoneNumbers.json    -d
 "PhoneNumber=5555" -d "VoiceUrl=http://127.0.0.1:8080/restcomm/demos/test.xml"
```

If the command is successful, you will see an output similar to the following:

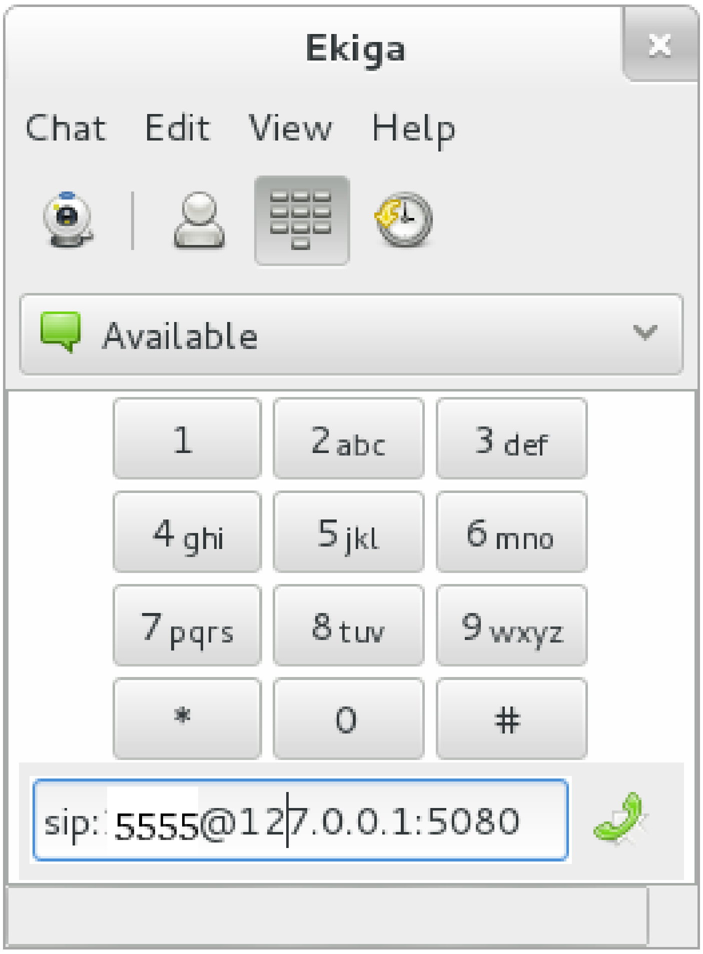
```
{
  "sid": "PN0834628c131e4b66bc862ae0b21b7cfa",
  "account_sid": "ACae6e420f425248d6a26948c17a9e2acf",
  "friendly_name": "5555",
  "phone_number": "+15555",
  "voice_url": "http://127.0.0.1:8080/restcomm/demos/hello-play.xml",
  "voice_method": "POST",
  "voice_fallback_method": "POST",
  "status_callback_method": "POST",
  "voice_caller_id_lookup": false,
  "date_created": "2013-08-15T17:45:21.409-06:00",
  "date_updated": "2013-08-15T17:45:21.409-06:00",
  "sms_method": "POST",
  "sms_fallback_method": "POST",
  "api_version": "2012-04-24",
               "uri":        "/restcomm/2012-04-24/Accounts/ACae6e420f425248d6a26948c17a9e2acf/
IncomingPhoneNumbers/PN0834628c131e4b66bc862ae0b21b7cfa.json"
```

Next, you have to configure your SIP phone and make a call to the `test.xml` dialing number 5555

Known Issues with some SIP Phones

Some SIP phones have been known to have codec problems and do not correctly render the application as desired. In this example, we shall be using Ekiga.

Start the SIP phone Ekiga (see below) and dial `5555@127.0.0.1:5080`. You will hear the content of the Say Verb in the `test.xml` file.



sip:5555@127.0.0.1:5080

7.1.2. Record Verb

Create a file called test.xml in the `$RESTCOMM_HOME/standalone/deployments/restcomm.war/demos/` directory and copy the content of the application below into the `test.xml` file and save it. Dial the number 5555 and when you hear the beep sound, leave a message and hangup when you are finished.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
        <Say>Welcome to RestComm, leave a message after the beep</Say>
 <Record maxLength="30"/>
</Response>
```

A wave file will be recorded and saved in the `$RESTCOMM_HOME/standalone/deployments/restcomm.war/recordings/` directory. You can use any media player to listen to the recorded voice message.

7.1.3. Dial Verb and Client

In order to use the Dial verb, you will need two users and register them to RestComm.

Restcomm has already two users created for you for testing purposes, alice and bob

Alice Restcomm client. Username = alice / Password = 1234

Bob Restcomm client. Username = bob / Password = 1234

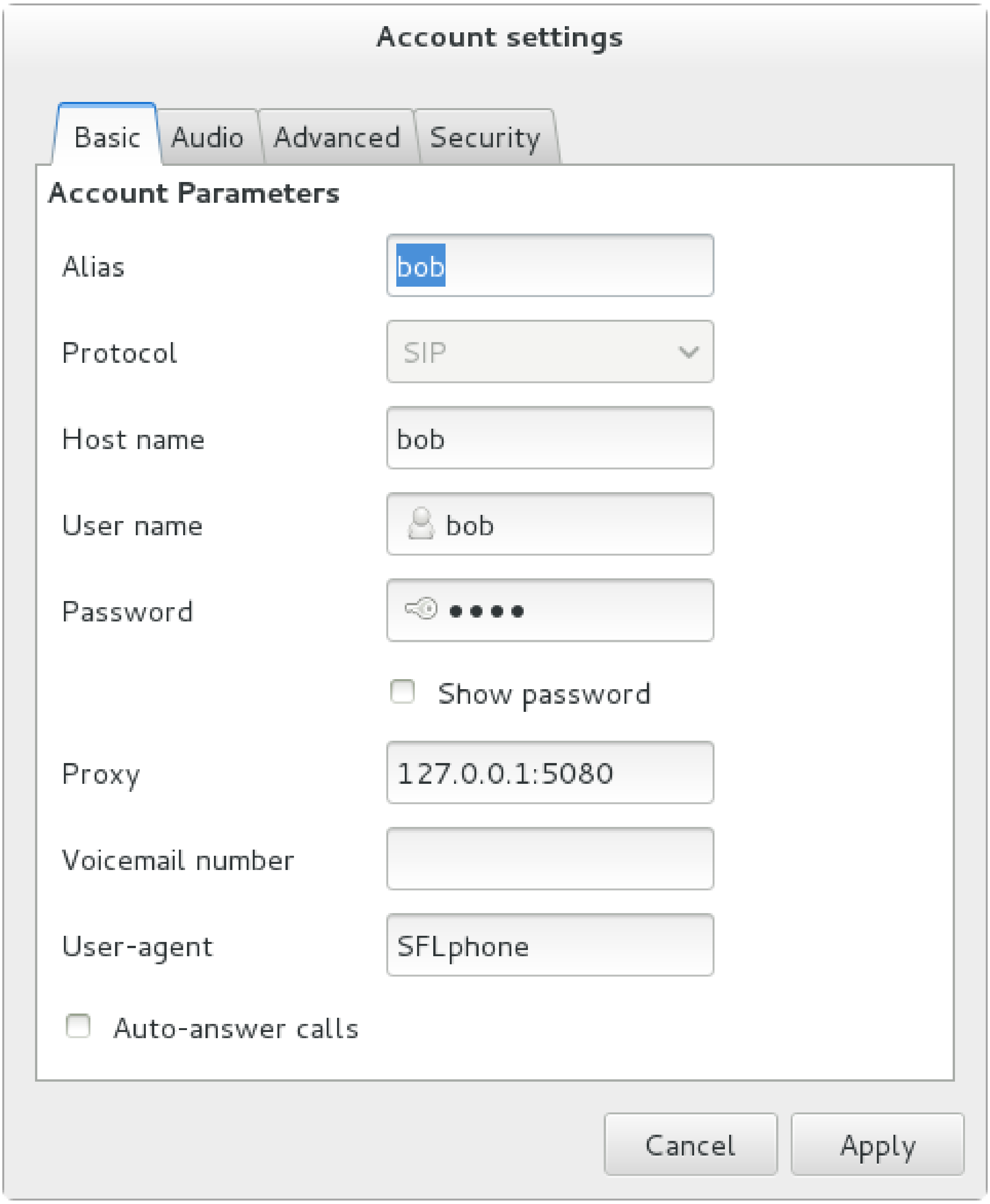
Register Alice and Bob using softphone Sflphone.

In order to start two instances of Sflphone on the same computer, you need to start the second instance using `sudo Sflphone`.

When Sflphone is started, go to the the menu Edit->Accounts, then press the Add button

Fill out the configuration for Bob as shown in the screenshot below:

Sflphone Basic Configuration.



Account settings

Basic | Audio | Advanced | Security

Account Parameters

Alias: bob

Protocol: SIP

Host name: bob

User name: bob

Password: ●●●●

□ Show password

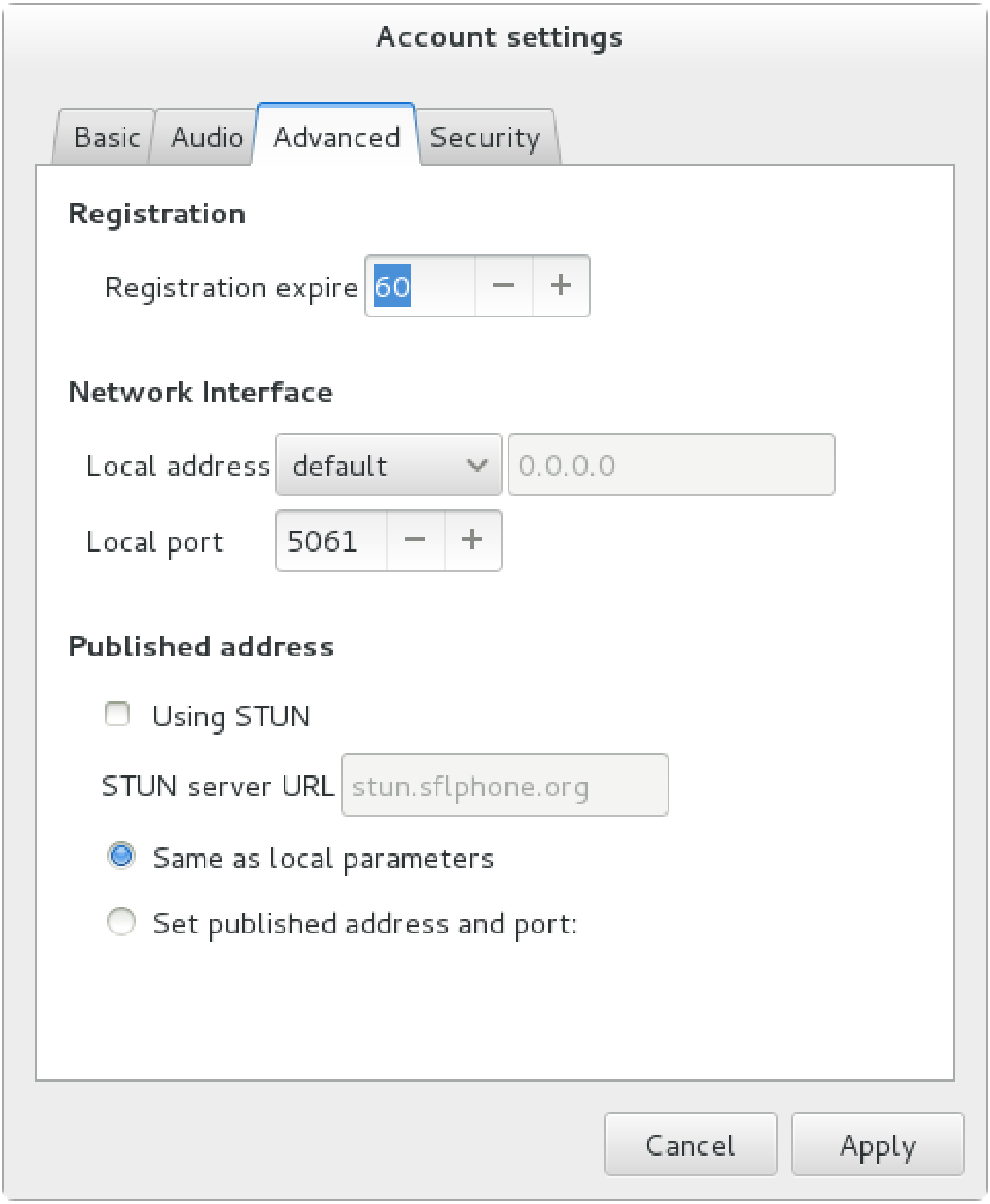
Proxy: 127.0.0.1:5080

Voicemail number:

User-agent: SFLphone

□ Auto-answer calls

Sflphone Advanced Configuration. Make sure the port number is set to 5061



Account settings

Basic | Audio | Advanced | Security

Registration

Registration expire: 60

Network Interface

Local address: default — 0.0.0.0

Local port: 5061

Published address

□ Using STUN

STUN server URL: stun.sflphone.org

● Same as local parameters

○ Set published address and port:

In the second instance of Sflphone, register user Alice following the same procedure used for Bob. In the Advanced settings, make sure the port number for user Alice is set to 5060

Copy the content of the application below into the test.xml file and save. .

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
        <Say>Welcome to RestComm, you are being forwarded to Alice</Say>
 <Dial callerid="+123456789">
 <Client>alice</Client>
 </Dial>
</Response>
```

From the Sflphone registered with user Bob, make a call to Alice. Enter the name `alice` and make the call

7.1.4. Dial Verb and Uri

Copy the Dial Uri application below to the `test.xml`. Remember that the port number for user Alice is 5060 and that is the port that is used in the application. If you want to use a different port, you also have to change the SIP Uri to reflect that. You can then make a call from user Bob to Alice.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
        <Say>Welcome to RestCom, you are being forward to Alice</Say>
 <Dial callerid="+1234567899">
 <Uri>sip:alice@127.0.0.1:5060</Uri>
 </Dial>

</Response>
```

7.1.5. Dial Verb and Conference

Copy the Dial Conference application below to the `test.xml` and save the file. You can dial any number like 4321 and RestComm will read the Say verb and make a beep sound when you join the conference room. You can use another SIP phone to join the same conference. It works better if you are on a different computer as this reduces the echo generated from the microphones.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
        <Say>Welcome to RestCom, you are now joining the conference room</Say>
 <Dial>
 <Conference>test-conference</Conference>
 </Dial>
</Response>
```

7.1.6. Gather Verb

This verb is used to get user input and instruct RestComm to perform a specific task. This example is a little bit more elaborate and it will require the creation of a php file. You also must host the php file on a web server like Apache. Copy the Gather application below into the `test.xml` and save the file.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <Gather action="http://127.0.0.1/test-user-input.php" numDigits="1">
   <Say>Welcome to Telestax RestCom.</Say>
   <Say>For opening hours, press 1.</Say>
   <Say>to talk to Alice, press 2.</Say>
  </Gather>
 <!-- If customer doesn't input anything, prompt and try again. -->
  <Say>Sorry, I didn't get your response.</Say>
  <Redirect>http://127.0.0.1:8080/restcomm/demos/test.xml</Redirect>
 </Response>
```

Create a php file in the Apache /var/www/html directory.

You can use any web server of your choice, in this example, we shall be using Apache on a Linux computer

Start the httpd server as follows `sudo service httpd start`

Create a new file called `test-user-input.php` in the /var/www/html directory

Copy the php application below into the `test-user-input.php` and save

```
<?php
    header('Content-type: text/xml');
    echo '<?xml version="1.0" encoding="UTF-8"?>';

    echo '<Response>';

    $user_input = (int) ($_POST['Digits']);

    if ($user_input == 1)

    {
         echo '<Say>Our Opening hours are 24 hours 7 Days a week </Say>';
        echo '<Say>Telestax appreciates your business</Say>';
        echo '<Say>You may Hang up or wait to be redirected to the main menu</Say>';
        echo '<Redirect>http://127.0.0.1:8080/restcomm/demos/test.xml</Redirect>';

    }  elseif($user_input == 2) {

        echo '<Say>You are being forwarded to Alice</Say>';
        echo '<Dial callerid="+1234568789">';
        echo '<Client>alice</Client>';
 echo '</Dial>';

     }
    echo '</Response>';
?>
```

This example welcomes the user and offers two options. If the user presses 1, he hears the openining hours message. If the user presses 2, he will be redirected to user Alice.

To test the application using the Gather verb, dial any number from user Bob and follow the application instruction.

Using the Fax, SMS and other features of RestComm

Please note that Telestax also offers TelScale RestComm hosted on Amazon AWS. If you want to learn more about how to use DID provisioning and more, please visit www.telestax.com [http://www.telestax.com]