2. Overview of sipXcom APIs

This is a technical manual describing the APIs used by the sipXcom/Uniteme software. It contains information for developers and system administrators on installing, managing and developing for the sipXcom/Uniteme software.

2.1 About sipXcom APIs

At the heart of the eZuce Uniteme Communications Platform are our extensive collection of Web Service APIs. By exposing the power of the platform through APIs customers can customize their users' communications experience within their organization.

Because Uniteme can act as a go-between for aging systems and new customer applications, sipXcom APIs acts as a link to assist communication or integration that allows developers to build a single application that interacts with many different systems. In support of this functionality, several different web service protocols and authentication schemes are available.

Uniteme is built with a variety of software languages a number of embedded open source applications. For more information, see the *eZuce Uniteme Third Party Component List*.

Benefits of using Web APIs

1. Integration with other systems. For example, integration with a given company's internal user management system.
2. Integration with other applications. For example, Microsoft Outlook© toolbar add-ons.
3. Internally inside Uniteme when two services on potentially two different servers to communicate with each other.

What are Uniteme Web APIs

Uniteme utilizes SOAP and REST API's. SipXconfig exposes web services and allows clients to connect and to manage, communicate and diagnose data.
REST and SOAP are two methods of exposing web services.
See below some characteristics of the two methods:

- REST web services (RESTful API) do not require XML-based web service protocols (SOAP and WSDL) to support their interfaces. This means that the client doesn't need any contract to call such web service and are described by URL. Each unique URL is a representation of some object. For instance, calling {+}https://localhost/sipxconfig/rest/my/conferences+ returns all the conferences for that user and calling {+}https://localhost/sipxconfig/rest/my/contact-information+ retrieves contact information for the user.
- SOAP (SOAP API) is an older and heavier protocol from a performance and development perspective. Clients that connect to this web service should respect a contract written in a description language file published by the server (For example, WSDL: {+}https://localhost/sipxconfig/services/UserService?wsdl+).

RESTlet is the framework used by sipXconfig for exposing RESTful API. For exposing SOAP API, eZuce utilities Apache Axis framework.
Earlier in the system development of Uniteme/sipXcom sipXconfig (the configuration service) started by providing SOAP web services but as REST technology evolved it was eventually adopted and further SOAP development has been put on hold.

2.1.1 REST APIs

About REST APIs
REST (acronym for "REpresentational State Transfer") represents a new approach to systems architecture and a lightweight alternative to web services. eZuce Inc provides a RESTful interface through which you can retrieve information about an instance and make configuration changes. Using the REST interface's simple HTTP calls, you can configure Uniteme and Reach. Requests on resources are implemented with the standard HTTP methods:

- **GET** to read;
- **PUT** to create;
- **POST** to update;
- **DELETE** to delete.

The sipXcom REST API is served over HTTPS. To ensure data privacy, unencrypted HTTP is not supported.
Base URL
The base URL for the Configuration API is the following: {+}https://host.example.com/sipxconfig/rest+
Using REST Services with cURL
Here is an example of how to print the content of a phonebook named *Sales* to standard output in CSV format:

```
curl --insecure --basic -u superadmin [https://]{host}/sipxconfig/rest/phonebook/sales
```

To test the call placing service with cURL use:

```
curl --insecure -X PUT [https://]{user}:{password}@{host}/sipxconfig/rest/call/{number}
```

HTTP **PUT** to service URL determines sipXconfig to place call to *{number}*. The call is placed using authorized user credentials. It works in the same way as click-to-call functionality available in the User Portal. User's phone rings first and when answered, it places the call to *{number}*.
To test the call forwarding service with cURL use:

curl --insecure -X PUT [https://]{user}:{password}@{host}/sipxconfig/rest/my/forward/ --data-binary @{file.xml}

HTTP **PUT** to service URL creates a call forwarding scheme depending on the XML content. The same result can be obtained through the User Portal, using the call forwarding page. Specified numbers will be called in cascade using the specified configuration:

```
<call-sequence>
<rings>
<ring>
<expiration>20</expiration>
<type>If no response</type>
<enabled>true</enabled>
<number>100</number>
</ring>
</rings>
<withVoicemail>false</withVoicemail>
</call-sequence>
```

2.1.2 SOAP APIs

About SOAP APIs
The SOAP API for sipXconfig lets you perform many operations offered by sipXconfig in a programmatic way and without interacting with the sipXconfig Web user interface. The sipXconfig API is continuously extended and offers most of the functionality available in the web interface.
Base URL
The base URL for the Configuration API is the following:
https://<host>/sipxconfig/services
Using SOAP Services
Benefits of using SOAP APIs:

- You can integrate the sipXcom functionality with your company's IT infrastructure (intranet).
- You can automate or script processes such as: adding or importing users, updating phones, assigning phones to groups and others.
- You can customize the sipXconfig User Interface (UI) and to suit your needs.

You can use SOAP with WSDL, which is a formal API definition, and generate bindings in your preferred programming language (Python, Perl, Ruby, Java, and others). It is recommended to select a programming language with good SOAP client support.

Resources for building SOAP web-based service clients
Every sipXconfig installation already publishes the SOAP API on URLs https://yourdomain/sipxconfig/services/*Service.
Ruby
From the WSDL, you can use the SOAP4R project to build client bindings.
Perl
Install SOAP for Perl
perl -MCPAN -e 'install SOAP::Lite'
Then you can adapt the sample script to your needs. For more information, see {+}http://www.perl.org/+.
Command Line
Use the following command line for SOAP web services:

```
java -jar $WsdlDocDir/wsdldoc.jar {color}
-title "sipXconfig SOAP API v3.2" {color}
-dir `pwd`"/ws-api-3.2" {color}
http://sipXcom.sipfoundry.org/rep/sipXcom/main/sipXconfig/web/src/org/sipfoundry/sipxconfig/api/sipxconfig.wsdl
```

2.1.3 OpenFire APIs

Openfire is a cross-platform real-time collaboration server based on the XMPP (Jabber) protocol. For more information, see http: *//www.igniterealtime.org/.*

2.1 Using sipXcom Web APIs

Areas of extensibility

1. The available group web service APIs have been categorized according to their usage:

To do this...	Use this API...
Configure the system, users, devices and features.	a. Manage
Control communications in the system.	a. Communicate

Retrieve information about the system.	a. Diagnose

Supported languages

The sipXcom Web APIs support development written in any language that supports SOAP and REST For example: JSON, XML, C++, Java, Ruby and others.

Compilers and software required

You can write using any standard word processing application and compile your programs using Visual C+, Visual Studio, .Net, Unix C, Unix C+, C# compilers. **HTTP response status codes**
The following table lists the generic HTTP response status codes for the **GET** (retrieve), **POST** (create), **PUT** (modify), and **DELETE** (remove) methods of the sipXcom APIs:

Response Code	HTTP Request	Description
200 OK	GET, PUT, DELETE	Request successful
400 Bad Request	GET, POST, PUT, DELETE	Bad query
401 Unauthorized	GET, POST, PUT, DELETE	Request requires user authentication
403 Forbidden	GET, POST, PUT, DELETE	Authentication failure
404 Not Found	GET, POST, PUT, DELETE	Resource not found
408 Request Timeout	GET, POST, PUT, DELETE	Request has timed out
500 Server Error	GET, POST, PUT, DELETE	Internal server error
502 Bad Gateway	GET, POST, PUT, DELETE	Data store failure

2.2. Adding new services

sipXconfig is implemented using the Restlet Framework API to add RESTful services. For more information on the Restlet Framework API, see _http://www.restlet.org._
Note: The RESTlet API version used by siXecs requires HTTP PUT request to have a body. You can add space, or anything to the body to work around it.
Service design
Follow these seps to design your REST service, prior to service implementation:

- Identify resources and design the resource structure
- Determine which operations (GET, PUT, POST, DELETE) will be available for which resources
- Map resources into URI structure
- Determine representation for each resource (specific format XML/JSON)

Service implementation
Follow these steps to implement a web service:

- Implement resources.

Each resource is an instance of RESTlet resource class. Every RESTful function is mapped into one or more resource methods. For example GET method is implemented by *Resource.represent*
Resource instances are managed by Spring. You can inject any context or manager that is necessary to perform service operations. For example phone book service delegates bulk of the work to PhonebookManager.

- Write unit test for resource and representation
- For resources verify XML/JSON representation generation and parsing
- Implement representations

sipXconfig services intended to be used pragmatically should at the minimum support XML and JSON representations. Read-only services marshal sipXconfig domain objects into representation.

- Write service system tests

Those are external tests written in Java/Ruby/Python that exercise the service. They are similar to UI tests.