



Small Medium Enterprise (SME)

IP Office 12.2
Management API Developer
Reference Guide

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

“Documentation” means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya may generally make available to users of its products and Cloud Services. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya’s agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked websites referenced within this site or documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on Avaya hardware and software. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya’s standard warranty language, as well as information regarding support for this product while under warranty is available to Avaya customers and other parties through the Avaya Support website: <https://support.avaya.com/helpcenter/getGenericDetails?detailId=C20091120112456651010> under the link “Warranty & Product Lifecycle” or such successor site as designated by Avaya. Please note that if You acquired the product(s) from an authorized Avaya Channel

Partner outside of the United States and Canada, the warranty is provided to You by said Avaya Channel Partner and not by Avaya.

“Cloud Service” means a cloud service subscription that You acquire from either Avaya or an authorized Avaya Channel Partner (as applicable) and which is described further in the applicable Service Description or other service description documentation regarding the applicable cloud service. If You purchase a Cloud Service subscription, the foregoing limited warranty may not apply but You may be entitled to support services in connection with the Cloud Service as described further in your service description documents for the applicable Cloud Service. Contact Avaya or Avaya Channel Partner (as applicable) for more information.

Cloud Service

THE FOLLOWING APPLIES IF YOU PURCHASE A CLOUD SERVICE SUBSCRIPTION FROM AVAYA OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE), THE TERMS OF USE FOR CLOUD SERVICES ARE AVAILABLE ON THE AVAYA WEBSITE,

<https://www.avaya.com/en/legal/license-terms/> UNDER THE LINK “Avaya Terms of Use for Cloud Services” OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, AND ARE APPLICABLE TO ANYONE WHO ACCESSES OR USES THE CLOUD SERVICE. BY ACCESSING OR USING THE CLOUD SERVICE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE DOING SO (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THE TERMS OF USE. IF YOU ARE ACCEPTING THE TERMS OF USE ON BEHALF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THESE TERMS OF USE. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO ACCEPT THESE TERMS OF USE, YOU MUST NOT ACCESS OR USE THE CLOUD SERVICE OR AUTHORIZE ANYONE TO ACCESS OR USE THE CLOUD SERVICE. YOUR USE OF THE CLOUD SERVICE SHALL BE LIMITED BY THE NUMBER AND TYPE OF LICENSES PURCHASED UNDER YOUR CONTRACT FOR THE CLOUD SERVICE, PROVIDED, HOWEVER, THAT FOR CERTAIN CLOUD SERVICES IF APPLICABLE, YOU MAY HAVE THE OPPORTUNITY TO USE FLEX LICENSES, WHICH WILL BE INVOICED ACCORDING TO ACTUAL USAGE ABOVE THE CONTRACT LICENSE LEVEL. CONTACT AVAYA OR AVAYA’S CHANNEL PARTNER FOR MORE INFORMATION ABOUT THE LICENSES FOR THE APPLICABLE CLOUD

SERVICE, THE AVAILABILITY OF ANY FLEX LICENSES (IF APPLICABLE), PRICING AND BILLING INFORMATION, AND OTHER IMPORTANT INFORMATION REGARDING THE CLOUD SERVICE.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, <https://www.avaya.com/en/legal/license-terms/>, UNDER THE LINK "AVAYA SOFTWARE LICENSE TERMS (Avaya Products)" OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA LLC, ANY AVAYA AFFILIATE, OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AVAYA CHANNEL PARTNER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA CHANNEL PARTNER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS "YOU" AND "END USER"), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA LLC OR THE APPLICABLE AVAYA AFFILIATE ("AVAYA").

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Cloud Service, or hardware provided by Avaya. All content on this site, the documentation, Cloud Service, and the product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and/or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

Virtualization

The following applies if the product is deployed on a virtual machine. Each product has its own ordering code and license types. Note that each Instance of a product must be separately licensed and ordered. For example, if the end user customer or Avaya Channel Partner would like to install two Instances of the same type of products, then two products of that type must be ordered.

Third Party Components

"Third Party Components" mean certain software programs or portions thereof included in the Software or Cloud Service may contain software (including open source software) distributed under third party agreements ("Third Party Components"), which contain terms regarding the rights to use certain portions of the Software ("Third Party Terms"). As required, information regarding distributed Linux OS source code (for those products that have distributed Linux OS source code) and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the products, Documentation or on Avaya's website at: <https://www.avaya.com/en/legal/third-party-terms/> or such successor site as designated by Avaya. The open source software license terms provided as Third Party Terms are consistent with the license rights granted in these Software License Terms, and may contain additional rights benefiting You, such as modification and distribution of the open source software. The Third Party Terms shall take precedence over these Software License Terms, solely with respect to the applicable Third Party Components to the extent that these Software License Terms impose greater restrictions on You than the applicable Third Party Terms.

The following applies if the H.264 (AVC) codec is distributed with the product. THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM VIA LICENSING ALLIANCE. SEE <https://www.via-la.com/>.

Service Provider

THE FOLLOWING APPLIES TO AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS OR SERVICES. THE PRODUCT OR CLOUD SERVICE MAY USE THIRD PARTY COMPONENTS SUBJECT TO THIRD PARTY TERMS AND REQUIRE A SERVICE PROVIDER TO BE INDEPENDENTLY LICENSED DIRECTLY FROM THE THIRD PARTY SUPPLIER. AN AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS MUST BE AUTHORIZED IN WRITING BY AVAYA AND IF THOSE HOSTED PRODUCTS USE OR EMBED CERTAIN THIRD PARTY SOFTWARE, INCLUDING BUT NOT LIMITED TO MICROSOFT SOFTWARE OR CODECS, THE AVAYA CHANNEL PARTNER IS REQUIRED TO INDEPENDENTLY OBTAIN ANY APPLICABLE LICENSE AGREEMENTS, AT THE AVAYA CHANNEL PARTNER'S EXPENSE, DIRECTLY FROM THE APPLICABLE THIRD PARTY SUPPLIER.

WITH RESPECT TO CODECS, IF THE AVAYA CHANNEL PARTNER IS HOSTING ANY PRODUCTS THAT USE OR EMBED THE G.729 CODEC, H.264 CODEC, OR H.265 CODEC, THE AVAYA CHANNEL PARTNER ACKNOWLEDGES AND AGREES THE AVAYA CHANNEL PARTNER IS RESPONSIBLE FOR ANY AND ALL RELATED FEES AND/OR ROYALTIES. THE G.729 CODEC IS LICENSED BY Sangoma Technologies Corporation SEE <https://www.asterisk.org/products/add-ons/g729-codec/>. THE H.264 (AVC) CODEC IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO: (I) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (II) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION FOR H.264 (AVC) AND H.265 (HEVC) CODECS MAY BE OBTAINED FROM VIA LICENSING ALLIANCE. SEE <https://www.via-la.com/>.

Compliance with Laws

Customer acknowledges and agrees that it is responsible for complying with any applicable laws and regulations, including, but not limited to laws and regulations related to call recording, data privacy, intellectual property, trade secret, fraud, and music performance rights, in the country or territory where the Avaya product is used.

Preventing Toll Fraud

"Toll Fraud" is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of Toll Fraud associated with your system and that, if Toll Fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Toll Fraud intervention

If You suspect that You are being victimized by Toll Fraud and You need technical assistance or support, call Technical Service Centre Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support website: <https://support.avaya.com> or such successor site as designated by Avaya.

Security Vulnerabilities

Information about Avaya's security support policies can be found in the Security Policies and Support section of <https://support.avaya.com/security>. Suspected Avaya product security vulnerabilities are handled per the Avaya Product Security Support Flow (<https://support.avaya.com/css/P8/documents/100161515>).

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support website: <https://support.avaya.com>, or such successor site as designated by Avaya.

Contact Avaya Support

See the Avaya Support website: <https://support.avaya.com> for product or Cloud Service notices and articles, or to report a problem with your Avaya product or Cloud Service. For a list of support telephone numbers and contact addresses, go to the Avaya Support website: <https://support.avaya.com> (or such successor site as designated by Avaya), scroll to the bottom of the page, and select Contact Avaya Support

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.2.1	REST Web Services	1
1.2.2	Operation Return Codes	2
1.3	Terminology and Acronyms	3
1.4	Change History	3
2	Web Services in IP Office	4
2.1	Accessing Management API for IPOL solution	4
2.2	Accessing Management API for IP500 V2 solution	5
2.3	Management API User	5
2.3.1	Creating Management API service user using Web Manager	5
2.3.2	Creating Management API service user using IP Office Manager	6
2.4	Data Format	7
2.5	Error Responses	7
2.6	Successful Response	8
2.7	Session Services	9
2.8	Administration/Configuration Services (REST)	12
2.8.1	ProvisionIPAddress	14
2.8.2	Systems	17
2.8.3	Licenses	28
2.8.4	Users	33
2.8.5	Extensions	57
2.8.6	HuntGroups	73
2.8.7	ShortCodes	90
2.8.8	Incoming Call Routes (ICR)	104
2.8.9	Incoming Line Group Ids	117
2.8.10	Outgoing Line Group Ids	119
3	Appendix A – XML Schemas	122
3.1	Common XSDs	122
3.1.1	DATA	122
3.1.2	WS_OBJECT	122
3.1.3	RESPONSE	123
3.1.4	REFERENCE	123
3.1.5	ERROR	124
3.1.6	WARNING	124
3.2	SYSTEM	125
3.3	LICENSE	129
3.4	USER	130
3.5	EXTENSION	131
3.6	HUNTGROUPE	131
3.7	SHORTCODE	132
3.8	INCOMINGCALLROUTE (ICR)	132

3.9 LINE.....	133
3.10 INCOMINGGROUPIDS.....	134
3.11 OUTGOINGGROUPIDS.....	135
4 Appendix B – Understanding the Data	135
4.1 ProvisionIPAddress	135
4.2 System.....	135
4.3 Licenses.....	138
4.4 Users.....	143
4.5 Extensions	148
4.6 HuntGroups	150
4.7 ShortCodes	151
4.8 Incoming Call Routes.....	152
4.9 Incoming Line Group Ids	153
4.10 Outgoing Line Group Ids	153
5 Appendix C – Common code snippets	154
5.1 Sample code for GET method.....	154
5.2 Sample code for PUT method	155
5.3 Sample code for POST method.....	156
5.4 Sample code for DELETE method.....	157
5.5 Certificate validation for .NET.....	159

1 Introduction

1.1 Purpose

This document is designed with two objectives in mind:

- Document the REST based Management APIs in IP Office Platform.
- Developers utilizing or wish to understand the Management IP Office APIs

As such, this is a highly technical document intended for the development community.

1.2 Scope

Web Services are typically referred to as Application Programming Interfaces (API) that are accessed via the web (or over a network) and which provide capabilities or services offered by a remote host.

Accesses to web services APIs is provided using the HTTPS protocol and in REST format.

1.2.1 REST Web Services

REST – Representational State Transfer is an architectural style for accessing information on the web. This is becoming increasingly popular and an easier way to access services on the web. In the REST architecture style, information that is available on the server side is considered as resources and these resources can be accessed in a uniform way using web – Uniform Resource Identifier (URI) and the web protocol, HTTP. Since HTTP is preferred as the protocol for communication, REST constraints the architecture to client-server architecture which is stateless in nature. Resources on the web are identified by URIs. Web Services that are created and accessed using REST principles are called RESTful Web Services.

The REST request consists of a URI with parameters. When the IP Office receives a REST request, the Web Service component determines the operation type parses the parameters and performs the requested task. The output of a REST request is either an HTTP error status or an HTTP status of 200 OK with XML or JSON formatted results’.

As REST is built on top of HTTP 1.1, the following table presents a mapping of HTTP standard operations and the corresponding standard REST behaviors.

HTTP Method	REST Meaning	Details
GET	Query details – result is in XML format	Returns a list or a specific instance of a resource. Parameters can be passed in the URL of the request
PUT	Updates an existing object	PUT parameters are contained within the body of the request in xml format
POST	Creates a new object	POST parameters are contained within the body of the request in xml format
DELETE	Deletes an object	Parameters can be passed in the URL of the request

1.2.2 Operation Return Codes

The HTTP return codes are an indication of HTTP message validation and server errors. It does not indicate the application status of the request. For example, a POST request can be sent with invalid data in the message body and still return a 200 OK HTTP return code. The application status result will be returned in the message body.

200 OK

The 200 (Ok) status code is returned to the web client due to a successful request.

201 Created

The 201 (Created) status code is returned to the web client when a new object is created.

400 Bad Request

The 400 (Bad Request) status code will be returned when the request is invalid due to invalid parameters.

401 Unauthorized

The 401 (Unauthorized) status code will be provided when the request requires user authentication.

404 Not Found

The 404 (Not Found) status code will be provided when the request specifies an object that doesn't exist.

405 Method Not Allowed

The request method specified for that resource is not supported by the server. The response will include a list of acceptable methods for that URI.

503 Service Unavailable

The server cannot fulfil the request at this time. The client should retry the request after a short delay as indicated in the Retry-After header (nominally 5 seconds).

1.3 Terminology and Acronyms

API	Application Programming Interface	
GUID	Globally Unique Identifier	An identifier of an object (User, Extension, Line,) that is unique across the system.
SME	Small Medium Enterprise	Any organization with a staff headcount ranging from 2 to 250.
IP Office	Avaya IP Office	Advance Unified Communications solution for the SME market.
SIP	Session Initiated Protocol	One of the standards-based protocol used for communication in IP Office
TBC	To be confirmed	
IT	Information technology	
CLI	Command Line Interface	
REST	Representational State Transfer	
SOAP	Simple Object Access Protocol	
UMS	Unified Messaging Service	Voicemail Pro Unified Messaging Service.
XML	Extensible Markup Language	
XSD	XML Schema Definition	
IPOL	Avaya IP Office for Linux	
SCN	Small Community Network	
URI	Uniform Resource Identifier	

1.4 Change History

Issue	Date	Details of changes
1.0	29-Feb-16	Issued initial document

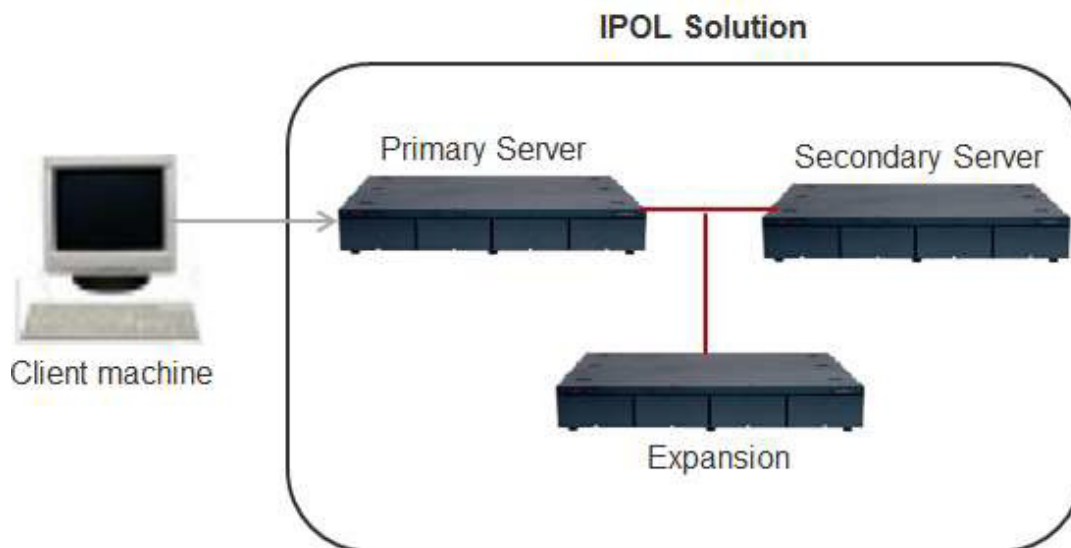
2.0	15-May-18	APIs updated for IP Office R11. Note that the R11 GA version of the API does not yet support SIP trunks - that will be added in a subsequent R11 Service Pack
3.0	1-August-25	APIs support for IP Office R12.2.

2 Web Services in IP Office

The IP Office Management API is a set of REST-based services which return results in XML or JSON.

To consume the Management API's a [Management API service user](#) must be created. Access to these services requires a session with the IP Office to be created by using the 'authenticate' REST service with an account and password with administrator privileges. When a session has been established, all other REST based Management APIs are available.

2.1 Accessing Management API for IPOL solution



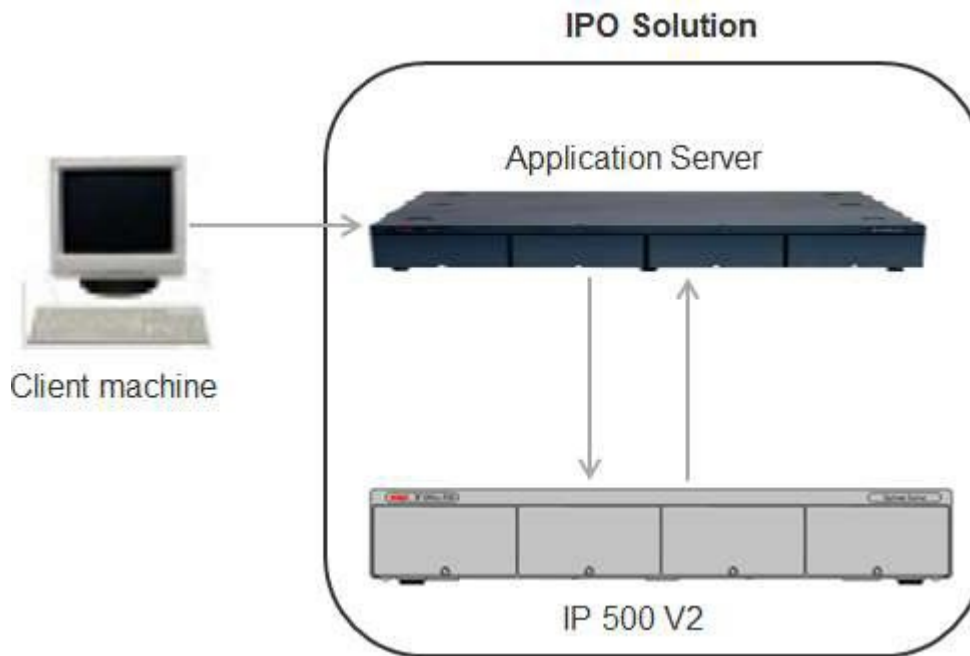
The above figure shows a typical IPOL solution. To access the Management APIs the Primary server will provide the consolidated web services for the entire solution. There will be no need to access the individual node. Authentication for the nodes will be done against the Primary and all nodes must have same



password for service user as the Primary node. If the passwords are different for the nodes, then the Primary node will fail to consolidate the object data and the data from the Secondary or Expansion will not be accessible from the Primary

node.

2.2 Accessing Management API for IP500 V2 solution



The above figure shows a typical IP 500 V2 solution for accessing the Management APIs. The Application Server will provide Web services for IP500 V2 Standard Edition. Authentication will be done against the Shell IPO security Database. Credentials of the service user on the nodes and Shell IPO must be the same. `ProvisionIPAddress` Web Service is provided to provision IP address of the IP 500 V2 standard edition so as to access the Management APIs. For SCN solution, the IP address of the IP500 V2 should be individually provisioned on the Application Server.

2.3 Management API User

In order to have access to the Management API, a service user must be configured with Management Service Group access on each system in the solution.

2.3.1 Creating Management API service user using Web Manager

Create a service user assigning the Rights Group Member "Management API Group"

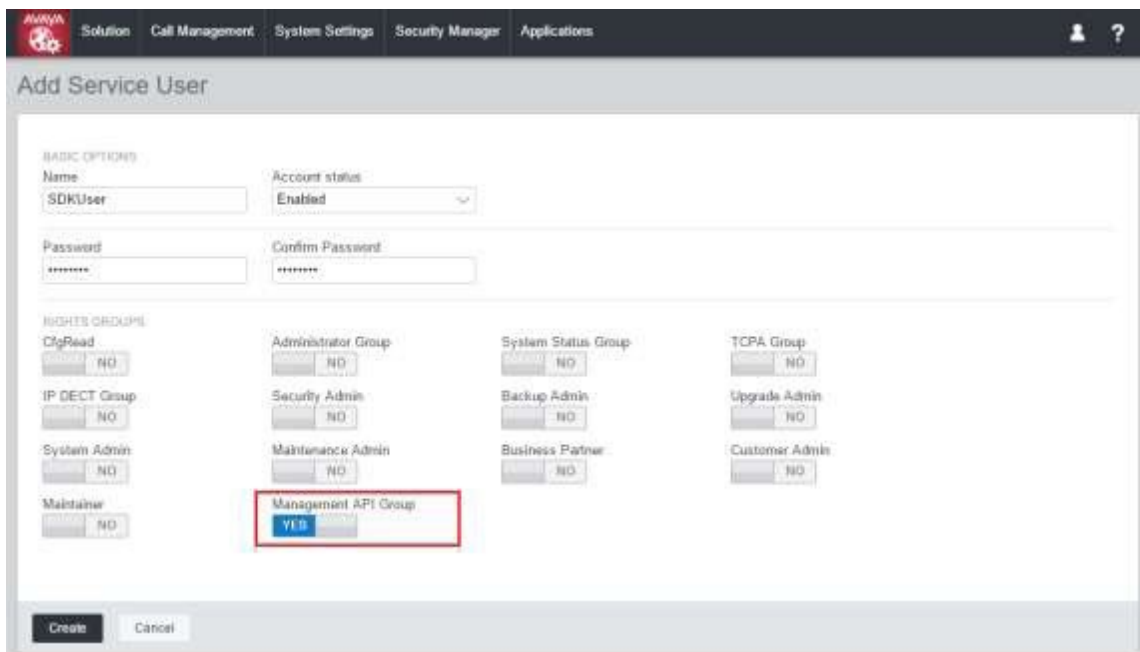
1. Login into Web Manager
2. Navigate through Security Manager->Service Users



3. Click “Add Service Users” on the “Service Users” page



4. Add a name and assign “Management API Group” for the service user.



2.3.2 Creating Management API service user using IP Office Manager

Create a service user assigning the Rights Group Member “Management API Group”



Management API Group Selection



Management API Group Rights

2.4 Data Format

The XSD files are provided with the APIs giving the definition of the XML data structure. Also for quick reference refer [XML Schemas](#) for further details.

The following sections provides the request and response structure. Note that many fields are enumerations, please refer the section [Understanding the data](#) for further detailed information about the enumeration values for the respective field value in the response.

2.5 Error Responses

Errors can occur at two levels in the API, at the HTTP layer where the request is not suitable for processing by the API and at the API level where the request is received by the API by the API but cannot process due to validation errors.

The API returns the error response in XML or JSON format with the response status of 0, and some more details on the error. HTTP error responses are covered in the section [Operation Return Codes](#). The API error return format is as follows:

XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <USER>
        <error>
          <error_code>9020</error_code>
          <error_desc>"A user or huntgroup with the specified Name Already
Exist."</error_desc>
          <cause>0</cause>
          <cause_data>ethel</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </USER>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": [
        {
          "@error_node": "true",
          "USER": {
            "error": [
              {
                "error_code": "9020",
                "error_desc": "\"A user or huntgroup with the specified Name Already
Exist.\"\"",
                "cause": "0",
                "cause_data": "ethel",
                "errorLocation": "10.240.0.5"
              }
            ]
          }
        }
      ]
    }
  }
}
```

2.6 Successful Response

A successful GET method for an API will return a body formatted in XML or JSON with response status of 1

XML Response

```
<response status="1">
  <data>
    <ws_object>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      }
    }
  }
}
```

2.7 Session Services

In order to begin using the API Administration and Configuration REST services, a session must be established with the IP Office. A session is created using the following REST API:

Method: GET

Description: REST service which creates a session with the IP Office.

URI: <https://{host}:7070/WebManagement/ws/sdk/security/authenticate>

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json.

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Sample code:

JavaScript

```
var xhr = new XMLHttpRequest();
xhr.open("GET",
  "https://{host}:7070/WebManagement/ws/sdk/security/authenticate", false);
xhr.setRequestHeader("X-User-Agent", "Avaya-SDKUser");
xhr.setRequestHeader("X-User-Client", "Avaya-WebAdmin");
xhr.setRequestHeader("Content-Type", "application/xml");
xhr.setRequestHeader("Authorization", "Basic " + btoa("username:password"));

xhr.send();
```


Java

```
try {
    URL url = new
URL("https://{host}:7070/WebManagement/ws/sdk/security/authenticate");
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setRequestMethod("GET");
    conn.setRequestProperty("X-User-Agent", "Avaya-SDKUser");
    conn.setRequestProperty("X-User-Client", "Avaya-WebAdmin");
    conn.setRequestProperty("Content-Type", "application/xml");
    String authStr = username+":"+password;
    byte[] bytesEncoded = Base64.encodeBase64(authStr.getBytes());
    String authEncoded = new String(bytesEncoded);
    conn.setRequestProperty("Authorization", "Basic "+authEncoded);

    if(conn.getResponseCode() != 200) {
        throw new RuntimeException("Failed : HTTP error code : "
        + conn.getResponseCode());
    }

    cookies = get_cookies(conn);

    BufferedReader br = new BufferedReader(new
InputStreamReader((conn.getInputStream())));

    String output;
    while ((output = br.readLine()) != null) {
        System.out.println(output);
    }
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
```

.NET

```
ServicePointManager.ServerCertificateValidationCallback = new
System.Net.Security.RemoteCertificateValidationCallback(CertificateValidationCallback);

String url = "https://{primaryhost}:7070/WebManagement/ws/sdk/security/authenticate";
CookieContainer cookieContainer = new CookieContainer();
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
request.Method = "GET";
request.ContentType = "application/xml";
request.Headers.Add("X-User-Client", "Avaya-WebAdmin");
request.Headers.Add("X-User-Agent", "Avaya-SDKUser");
String credentials = Convert.ToBase64String(ASCIIEncoding.ASCII.GetBytes(username + ":" +
password));
request.Headers.Add("Authorization", "Basic " + credentials);
request.CookieContainer = cookieContainer;

HttpWebResponse response = (HttpWebResponse)request.GetResponse();
var resStream = response.GetResponseStream();
var reader = new StreamReader(resStream);
String responseContent = reader.ReadToEnd();

response.Close();
```

Result:

```

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Set-Cookie: JSESSIONID=24121E0F893E7B8A97CACF6A6DEB1469; Path=/WebManagement/;
Secure; HttpOnly
Expires: Fri, 20 Nov 2015 12:59:47 GMT
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Fri, 20 Nov 2015 12:59:45 GMT

```

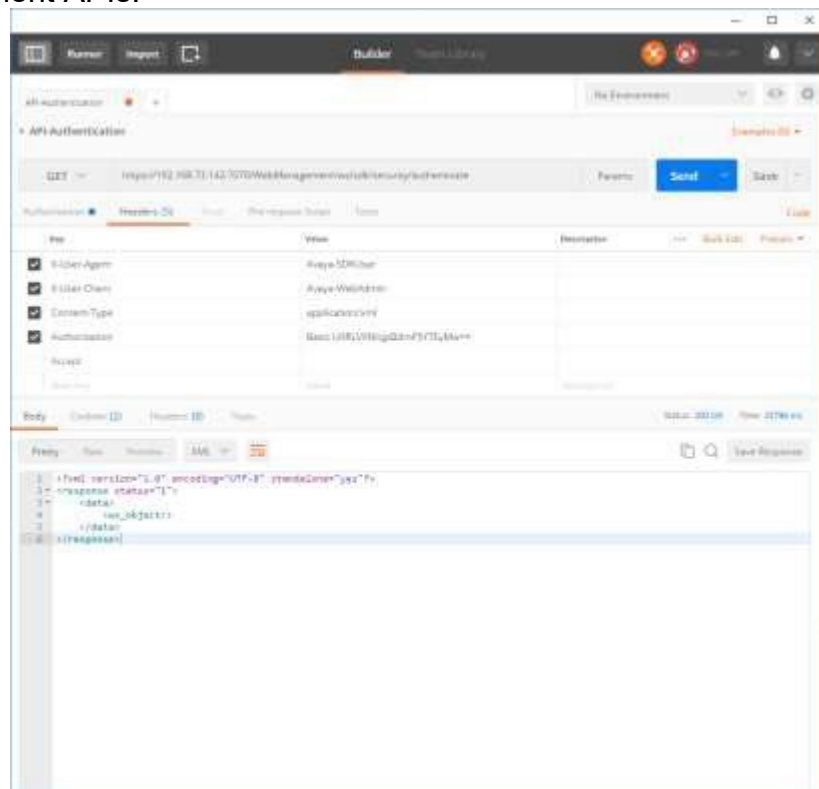
```

<?xml version="1.0" encoding="UTF8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object/>
  </data>
</response>

```

NOTE: On a successful request, the response will contain a ‘Set-Cookie’ header. Part of this header data is a JSESSIONID and SDK-Token which is the session id to be used in all subsequent REST requests for this session.

Alternatively a tool like POSTman Add-in can be used. Following is a sample to access the authenticate request. Similar way can be followed for all the Management APIs.



2.8 Administration/Configuration Services (REST)

Web Services defined in this section for administration/configuration of the system are all REST-based services.

Services described implement a number of administration/configuration functions related primarily to users, extensions, hunt groups and sip trunks.

Port: 7070

Base URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1>

URI	Parameters	Method	Description
<i>ProvisionIPAddress</i>	<i>ipaddress=<LAN 1 IP address of application server></i>	GET	Returns the IP Address of the provisioned IP500V2 system.
<i>ProvisionIPAddress</i>	<i>ipaddress=<LAN 1 IP address of application server></i>	PUT	Update the IP Address of the provisioned IP500V2 system.
<i>Systems</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Returns information on the nodes in a solution.
<i>Systems</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	PUT	Update the information on a solution node.
<i>Licenses</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Retrieves the licenses for the specified system.
<i>Users</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Returns a list of users defined on the system.
<i>Users</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	POST	Add a User to a System
<i>Users</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	PUT	Updates an existing user
<i>Users</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	DELETE	Deletes an existing user on a system.
<i>Extensions</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Returns a list of extensions defined on the system.
<i>Extensions</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	POST	Adds a new extension to a system.

<i>Extensions</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	PUT	Updates an existing extension on a system.
<i>Extensions</i>	<i>ipaddress=<LAN 1 IP address of solution node> guid=<GUID of extension to delete></i>	DELETE	Deletes an existing extension on a system.
<i>Huntgroups</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Returns a list of huntgroups defined on a system.
<i>Huntgroups</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	POST	Add a huntgroup to a system.
<i>Huntgroups</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	PUT	Update an existing huntgroup on a system.
<i>Huntgroups</i>	<i>ipaddress=<LAN 1 IP address of solution node> guid=<GUID of huntgroup to delete></i>	DELETE	Deletes an existing huntgroup on a system.
<i>Siplines</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Gets all SipLines defined on a system.
<i>Siplines</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	POST	Adds a new SipLine to a system.
<i>Siplines</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	PUT	Updates an existing SipLine on a system.
<i>Siplines</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	DELETE	Deletes an existing SipLine on a system.
<i>Shortcodes</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Gets all shortcodes defined on a system.
<i>Shortcodes</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	PUT	Updates an existing shortcode on a system.
<i>Shortcodes</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	POST	Adds a new short code to a system.
<i>Shortcodes</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	DELETE	Deletes an existing short code from a system.

<i>incoming_call_routes</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Gets all incoming call routes defined on a system.
<i>incoming_call_routes</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	PUT	Updates an existing incoming call route on a system.
<i>incoming_call_routes</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	POST	Adds a new incoming call route to a system.
<i>incoming_call_routes</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	DELETE	Deletes an existing incoming call route on a system.
<i>outgoing_group_ids</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Gets all outgoing line group ids defined on a system.
<i>incoming_group_ids</i>	<i>ipaddress=<LAN 1 IP address of solution node></i>	GET	Gets all incoming line group ids defined on a system.

2.8.1 ProvisionIPAddress

2.8.1.1 ProvisionIPAddress (GET)

Method: GET

Description: REST web service to get the Provisioned IP Address of IP500 V2 system in the solution.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/provisionIPAddress>

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded

Content-Type: application/xml OR application/json.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/provisionIPAddress>

Output Parameters:

An XML document containing information on each system defined in the solution or solution node including (but not limited to):

- Response Status – A value of 1 indicates Success, 0 is a failure
- IP Address of the provisioned IP500 V2 system.

XSDs: RESPONSE, DATA, WS_OBJECT, SYSTEM, ERROR

Example 1: Get the IP500V2 system provisioned IP Address defined in the solution.

```
GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/provisionIPAddress
HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 08 Dec 2015 15:58:53 GMT
Content-Type: application/xml
Content-Length: 8025
Date: Tue, 08 Dec 2015 15:58:51 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <ProvisionIPAddress>
        <IPAddress>192.168.42.6</IPAddress>
      </ProvisionIPAddress>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": {
        "ProvisionIPAddress": {
          "IPAddress": "192.168.42.6"
        }
      }
    }
  }
}
```

2.8.1.2 ProvisionIPAddress (PUT)

Method: PUT

Description: REST service which queries/lists systems in a solution or node in a solution. This web service is used to provision IP500 V2 IP Address using Application Server so as to use the Management APIs for IP500 V2.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/provisionIPAddress>

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Cookie: JSESSIONID=<session id obtained from authenticate>

Content-Type: application/xml OR application/json.

Sample code:

Refer section [Sample code for PUT](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/provisionIPAddress>

XML Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <ProvisionIPAddress>
        <IPAddress>192.168.42.6</IPAddress>
      </ProvisionIPAddress>
    </ws_object>
  </data>
</response>
```

JSON Request

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": {
        "ProvisionIPAddress": {
          "IPAddress": "192.168.42.6"
        }
      }
    }
  }
}
```

2.8.2 Systems

2.8.2.1 Systems (GET)

Method: GET

Description: REST service which queries/lists systems in a solution or node in a solution.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/systems>

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded

Input Parameters:

ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/systems>

Output Parameters:

An XML document containing information on each system defined in the solution or solution node including (but not limited to):

- Response Status – A value of 1 indicates Success, 0 is a failure
- Total Number of System Records
- Name – System name
- LANs
- License Server Information

XSDs: RESPONSE, DATA, WS_OBJECT, SYSTEM, ERROR

Example 1: Get all systems in the solution.

```
GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/systems HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 08 Dec 2015 15:58:53 GMT
Content-Type: application/xml
Content-Length: 8025
Date: Tue, 08 Dec 2015 15:58:51 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data set="75" totalRecords="1">
    <ws_object>
      <System>
        <Name>IPO-Primary</Name>
        <Password/>
        <MajorVersion>10</MajorVersion>
        <MinorVersion>1</MinorVersion>
        <Configuration>11</Configuration>
        <BuildVersion>45</BuildVersion>
        <Telephony>
          <Locale>eng</Locale>
          <ModemTxLevel>0</ModemTxLevel>
          <ModemCDThreshold>0</ModemCDThreshold>
          <ModemNoActivityTimeout>0</ModemNoActivityTimeout>
        </Telephony>
        <Voicemail>
          <Type>1</Type>
        </Voicemail>
        <LANs>
          <LAN id="LAN1">
```

```

<IPAddress>192.168.42.1</IPAddress>
<IPMask>255.255.255.0</IPMask>
<PrimaryTranslationIPAddress>0.0.0.0</PrimaryTranslationIPAddress>
<EnableNAT>false</EnableNAT>
<RIPMode>0</RIPMode>
<dhcp_address_number>81</dhcp_address_number>
<DHCPmode>4</DHCPmode>
<NetworkTopologyDiscovery>
  <StunServerAddress/>
  <ConnectivityType>7</ConnectivityType>
  <RefreshTime>0</RefreshTime>
  <PublicIPAddress>0.0.0.0</PublicIPAddress>
  <PublicPortUdp>0</PublicPortUdp>
  <PublicPortTcp>0</PublicPortTcp>
  <PublicPortTls>0</PublicPortTls>
  <STUNPort>3478</STUNPort>
  <RunOnFailure>false</RunOnFailure>
  <IsBehindFirewall>false</IsBehindFirewall>
</NetworkTopologyDiscovery>
<VoipGatekeeper>
  <H323GatekeeperEnable>true</H323GatekeeperEnable>
  <H323RemoteExtnEnable>false</H323RemoteExtnEnable>
  <H323RemoteExtnCallSignalPort>1720</H323RemoteExtnCallSignalPort>
  <H323SignalingOverTLS>0</H323SignalingOverTLS>
  <SipTrunksEnable>true</SipTrunksEnable>
  <SipRegistrarEnable>false</SipRegistrarEnable>
  <SipRemoteExtnEnable>false</SipRemoteExtnEnable>
  <DomainName/>
  <SipUdpEnable>true</SipUdpEnable>
  <UdpPort>5060</UdpPort>
  <RemoteUdpPort>5060</RemoteUdpPort>
  <SipTcpEnable>true</SipTcpEnable>
  <TcpPort>5060</TcpPort>
  <RemoteTcpPort>5060</RemoteTcpPort>
  <SipTlsEnable>false</SipTlsEnable>
  <TlsPort>5061</TlsPort>
  <RemoteTlsPort>5061</RemoteTlsPort>
  <ChallengeExpiryTime>10000</ChallengeExpiryTime>
  <RtpPortMin>40750</RtpPortMin>
  <RtpPortMax>50750</RtpPortMax>
  <RtpPortMinNat>40750</RtpPortMinNat>
  <RtpPortMaxNat>50750</RtpPortMaxNat>
  <RtcpMonitorEnable>true</RtcpMonitorEnable>
  <AutoCreateExtn>false</AutoCreateExtn>
  <AutoCreateUser>false</AutoCreateUser>
  <RemoteCallSignallingPort>1300</RemoteCallSignallingPort>
  <DHCPPrimarySSON>176</DHCPPrimarySSON>
  <DHCPSecondarySSON>242</DHCPSecondarySSON>
  <DHCPVlan>4294967295</DHCPVlan>
  <DHCPVlanSSON>232</DHCPVlanSSON>
  <DHCPVlanIDs/>
  <RTCPCollectorIpAddress>0.0.0.0</RTCPCollectorIpAddress>
  <SIPRegistrarFQDN/>
  <SipRegistrarAutoCreateExtnOrUser>false</SipRegistrarAutoCreateExtnOrUser>
</VoipGatekeeper>
<LanDHCPPools>
  <ApplyToAvayaIPPhonesOnly>false</ApplyToAvayaIPPhonesOnly>
  <DHCPPools>
    <DHCPPool id="0">
      <DHCPStartAddress>148.147.206.174</DHCPStartAddress>
      <DHCPSubnetMask>255.255.255.0</DHCPSubnetMask>
      <DHCPDefaultRouter>0.0.0.0</DHCPDefaultRouter>
      <DHCPPoolSize>81</DHCPPoolSize>
    </DHCPPool>
  </DHCPPools>
</LanDHCPPools>
<Scope>0</Scope>
<InitialKeepalives>false</InitialKeepalives>
<PeriodicTimeout>0</PeriodicTimeout>
<DiffServSettings>

```

```

        <DSCP>184</DSCP>
        <VideoDSCP>184</VideoDSCP>
        <DSCPMask>252</DSCPMask>
        <SigDSCP>136</SigDSCP>
    </DiffServSettings>
    <FirewallProfile/>
</LAN>
<LAN id="LAN2">
    <IPAddress>0.0.0.0</IPAddress>
    <IPMask>0.0.0.0</IPMask>
    <PrimaryTranslationIPAddress>0.0.0.0</PrimaryTranslationIPAddress>
    <EnableNAT>false</EnableNAT>
    <RIPMode>0</RIPMode>
    <dhcp_address_number>0</dhcp_address_number>
    <DHCPmode>2</DHCPmode>
    <NetworkTopologyDiscovery>
        <StunServerAddress/>
        <ConnectivityType>7</ConnectivityType>
        <RefreshTime>0</RefreshTime>
        <PublicIPAddress>0.0.0.0</PublicIPAddress>
        <PublicPortUdp>0</PublicPortUdp>
        <PublicPortTcp>0</PublicPortTcp>
        <PublicPortTls>0</PublicPortTls>
        <STUNPort>3478</STUNPort>
        <RunOnFailure>false</RunOnFailure>
        <IsBehindFirewall>false</IsBehindFirewall>
    </NetworkTopologyDiscovery>
    <VoipGatekeeper>
        <H323GatekeeperEnable>true</H323GatekeeperEnable>
        <H323RemoteExtnEnable>false</H323RemoteExtnEnable>
        <H323RemoteExtnCallSignalPort>1720</H323RemoteExtnCallSignalPort>
        <H323SignalingOverTLS>0</H323SignalingOverTLS>
        <SipTrunksEnable>true</SipTrunksEnable>
        <SipRegistrarEnable>false</SipRegistrarEnable>
        <SipRemoteExtnEnable>false</SipRemoteExtnEnable>
        <DomainName/>
        <SipUdpEnable>true</SipUdpEnable>
        <UdpPort>5060</UdpPort>
        <RemoteUdpPort>5060</RemoteUdpPort>
        <SipTcpEnable>true</SipTcpEnable>
        <TcpPort>5060</TcpPort>
        <RemoteTcpPort>5060</RemoteTcpPort>
        <SipTlsEnable>false</SipTlsEnable>
        <TlsPort>5061</TlsPort>
        <RemoteTlsPort>5061</RemoteTlsPort>
        <ChallengeExpiryTime>10000</ChallengeExpiryTime>
        <RtpPortMin>40750</RtpPortMin>
        <RtpPortMax>50750</RtpPortMax>
        <RtpPortMinNat>40750</RtpPortMinNat>
        <RtpPortMaxNat>50750</RtpPortMaxNat>
        <RtcpMonitorEnable>true</RtcpMonitorEnable>
        <AutoCreateExtn>false</AutoCreateExtn>
        <AutoCreateUser>false</AutoCreateUser>
        <RemoteCallSignallingPort>1300</RemoteCallSignallingPort>
        <DHCPPrimarySSON>176</DHCPPrimarySSON>
        <DHCPSecondarySSON>242</DHCPSecondarySSON>
        <DHCPVlan>4294967295</DHCPVlan>
        <DHCPVlanSSON>232</DHCPVlanSSON>
        <DHCPVlanIDs/>
        <RTCPCollectorIpAddress>0.0.0.0</RTCPCollectorIpAddress>
        <SIPRegistrarFQDN/>
        <SipRegistrarAutoCreateExtnOrUser>false</SipRegistrarAutoCreateExtnOrUser>
    </VoipGatekeeper>
    <LanDHCPPools>
        <ApplyToAvayaIPPhonesOnly>false</ApplyToAvayaIPPhonesOnly>
        <DHCPPools>
            <DHCPPool id="0">
                <DHCPStartAddress>0.0.0.1</DHCPStartAddress>
                <DHCPSubnetMask>0.0.0.0</DHCPSubnetMask>
                <DHCPDefaultRouter>0.0.0.0</DHCPDefaultRouter>
            </DHCPPool>
        </DHCPPools>
    </LanDHCPPools>
</LAN id="LAN2">

```

```

        <DHCPPoolSize>0</DHCPPoolSize>
    </DHCPPool>
</DHCPPools>
</LanDHCPPools>
<Scope>0</Scope>
<InitialKeepalives>false</InitialKeepalives>
<PeriodicTimeout>0</PeriodicTimeout>
<DiffServSettings>
    <DSCP>184</DSCP>
    <VideoDSCP>184</VideoDSCP>
    <DSCPMask>252</DSCPMask>
    <SigDSCP>136</SigDSCP>
</DiffServSettings>
<FirewallProfile/>
</LAN>
</LANS>
<PBXLicense>
    <EnableRemoteServer>true</EnableRemoteServer>
    <EnableProxyViaPrimaryIPOLine>false</EnableProxyViaPrimaryIPOLine>
    <LicenseServerIPAddress>127.0.0.1</LicenseServerIPAddress>
    <LicenseServerURL/>
    <LicenseServerURLSecondary/>
    <LicenseServerURN/>
    <LicenseServerPort>52233</LicenseServerPort>
    <LicenseServerID>000C299B41C5-WS-Nilesh</LicenseServerID>
    <LicenseMode>4</LicenseMode>
    <LicenseGracePeriod>30</LicenseGracePeriod>
    <ReservedLicenses>
        <VMProRecordingsAdmins>0</VMProRecordingsAdmins>
        <WaveUsers>0</WaveUsers>
        <VMProTtsPro>0</VMProTtsPro>
        <VMProTtsWindows>0</VMProTtsWindows>
        <VMProTts3rdParty>0</VMProTts3rdParty>
        <CtiLinkPro>0</CtiLinkPro>
        <EmbeddedVMPorts>0</EmbeddedVMPorts>
        <VMProPorts>2</VMProPorts>
        <SipSessions>0</SipSessions>
        <ACCS>0</ACCS>
        <BranchSystem>0</BranchSystem>
        <SMTrunks>0</SMTrunks>
        <UMSWebServices>0</UMSWebServices>
        <MacSoftphones>0</MacSoftphones>
        <ThirdPartyRecorder>0</ThirdPartyRecorder>
        <ServerEdition>1</ServerEdition>
        <AvayaIPEndpoints>0</AvayaIPEndpoints>
        <ThirdPartyIPEndpoints>0</ThirdPartyIPEndpoints>
        <Receptionist>0</Receptionist>
        <BasicUser>0</BasicUser>
        <OfficeWorker>0</OfficeWorker>
        <Teleworker>0</Teleworker>
        <MobileUser>0</MobileUser>
        <PowerUser>0</PowerUser>
        <AvayaSoftphone>0</AvayaSoftphone>
        <WebCollaboration>0</WebCollaboration>
        <UniversalPRIAdditionalchannels>0</UniversalPRIAdditionalchannels>
        <IPSecTunneling>0</IPSecTunneling>
        <CentralizedEndpoints>0</CentralizedEndpoints>
        <EssentialEdition>0</EssentialEdition>
        <PreferredEdition>0</PreferredEdition>
    </ReservedLicenses>
    <LicensedVersion/>
    <PLDSHostId>387859159143</PLDSHostId>
    <PLDSFileStatus>0</PLDSFileStatus>
    <SelectLicensingStatus>1</SelectLicensingStatus>
    <GraceDongleSerialNumber/>
    <DongleSerialGraceDays>0</DongleSerialGraceDays>
    <GracePLDSHostId/>
    <LicensingBasicUser>true</LicensingBasicUser>
</PBXLicense>
</System>

```

```
</ws_object>
</data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "@set": "75",
      "@totalRecords": "1",
      "ws_object": {
        "System": {
          "Name": "WS-Nilesh",
          "Password": "",
          "MajorVersion": 10,
          "MinorVersion": 1,
          "Configuration": 11,
          "BuildVersion": 45,
          "Telephony": {
            "Locale": "eng",
            "ModemTxLevel": 0,
            "ModemCDThreshold": 0,
            "ModemNoActivityTimeout": 0
          },
          "Voicemail": {
            "Type": 1
          },
          "LANS": {
            "LAN": [{
              "@id": "LAN1",
              "IPAddress": "192.168.42.1",
              "IPMask": "255.255.255.0",
              "PrimaryTranslationIPAddress": "0.0.0.0",
              "EnableNAT": false,
              "RIPMode": 0,
              "dhcp_address_number": 81,
              "DHCPmode": 4,
              "NetworkTopologyDiscovery": {
                "StunServerAddress": "",
                "ConnectivityType": 7,
                "RefreshTime": 0,
                "PublicIPAddress": "0.0.0.0",
                "PublicPortUdp": 0,
                "PublicPortTcp": 0,
                "PublicPortTls": 0,
                "STUNPort": 3478,
                "RunOnFailure": false,
                "IsBehindFirewall": false
              },
              "VoipGatekeeper": {
                "H323GatekeeperEnable": true,
                "H323RemoteExtnEnable": false,
                "H323RemoteExtnCallSignalPort": 1720,
                "H323SignalingOverTLS": 0,
                "SipTrunksEnable": true,
                "SipRegistrarEnable": false,
                "SipRemoteExtnEnable": false,
                "DomainName": "",
                "SipUdpEnable": true,

```

```

    "UdpPort": 5060,
    "RemoteUdpPort": 5060,
    "SipTcpEnable": true,
    "TcpPort": 5060,
    "RemoteTcpPort": 5060,
    "SipTlsEnable": false,
    "TlsPort": 5061,
    "RemoteTlsPort": 5061,
    "ChallengeExpiryTime": 10000,
    "RtpPortMin": 40750,
    "RtpPortMax": 50750,
    "RtpPortMinNat": 40750,
    "RtpPortMaxNat": 50750,
    "RtcpMonitorEnable": true,
    "AutoCreateExtn": false,
    "AutoCreateUser": false,
    "RemoteCallSignallingPort": 1300,
    "DHCPPPrimarySSON": 176,
    "DHCPSecondarySSON": 242,
    "DHCPVlan": 4294967295,
    "DHCPVlanSSON": 232,
    "DHCPVlanIDs": "",
    "RTCPCollectorIpAddress": "0.0.0.0",
    "SIPRegistrarFQDN": "",
    "SipRegistrarAutoCreateExtnOrUser": false
  },
  "LanDHCPPools": {
    "ApplyToAvayaIPPhonesOnly": false,
    "DHCPPools": {
      "DHCPPool": {
        "@id": "0",
        "DHCPStartAddress": "148.147.206.174",
        "DHCPSubnetMask": "255.255.255.0",
        "DHCPDefaultRouter": "0.0.0.0",
        "DHCPPoolSize": 81
      }
    }
  },
  "Scope": 0,
  "InitialKeepalives": false,
  "PeriodicTimeout": 0,
  "DiffServSettings": {
    "DSCP": 184,
    "VideoDSCP": 184,
    "DSCPMask": 252,
    "SigDSCP": 136
  },
  "FirewallProfile": ""
}, {
  "@id": "LAN2",
  "IPAddress": "0.0.0.0",
  "IPMask": "0.0.0.0",
  "PrimaryTranslationIPAddress": "0.0.0.0",
  "EnableNAT": false,
  "RIPMode": 0,
  "dhcp_address_number": 0,
  "DHCPmode": 2,
  "NetworkTopologyDiscovery": {
    "StunServerAddress": "",
    "ConnectivityType": 7,
    "RefreshTime": 0,

```

```

    "PublicIPAddress": "0.0.0.0",
    "PublicPortUdp": 0,
    "PublicPortTcp": 0,
    "PublicPortTls": 0,
    "STUNPort": 3478,
    "RunOnFailure": false,
    "IsBehindFirewall": false
  },
  "VoipGatekeeper": {
    "H323GatekeeperEnable": true,
    "H323RemoteExtnEnable": false,
    "H323RemoteExtnCallSignalPort": 1720,
    "H323SignalingOverTLS": 0,
    "SipTrunksEnable": true,
    "SipRegistrarEnable": false,
    "SipRemoteExtnEnable": false,
    "DomainName": "",
    "SipUdpEnable": true,
    "UdpPort": 5060,
    "RemoteUdpPort": 5060,
    "SipTcpEnable": true,
    "TcpPort": 5060,
    "RemoteTcpPort": 5060,
    "SipTlsEnable": false,
    "TlsPort": 5061,
    "RemoteTlsPort": 5061,
    "ChallengeExpiryTime": 10000,
    "RtpPortMin": 40750,
    "RtpPortMax": 50750,
    "RtpPortMinNat": 40750,
    "RtpPortMaxNat": 50750,
    "RtcpMonitorEnable": true,
    "AutoCreateExtn": false,
    "AutoCreateUser": false,
    "RemoteCallSignallingPort": 1300,
    "DHCPPPrimarySSON": 176,
    "DHCPSecondarySSON": 242,
    "DHCPVlan": 4294967295,
    "DHCPVlanSSON": 232,
    "DHCPVlanIDs": "",
    "RTCPCollectorIpAddress": "0.0.0.0",
    "SIPRegistrarFQDN": "",
    "SipRegistrarAutoCreateExtnOrUser": false
  },
  "LanDHCPPools": {
    "ApplyToAvayaIPPhonesOnly": false,
    "DHCPPools": {
      "DHCPPool": {
        "@id": "0",
        "DHCPStartAddress": "0.0.0.1",
        "DHCPSubnetMask": "0.0.0.0",
        "DHCPDefaultRouter": "0.0.0.0",
        "DHCPPoolSize": 0
      }
    }
  },
  "Scope": 0,
  "InitialKeepalives": false,
  "PeriodicTimeout": 0,
  "DiffServSettings": {
    "DSCP": 184,

```

```

        "VideoDSCP": 184,
        "DSCPMask": 252,
        "SigDSCP": 136
    },
    "FirewallProfile": ""
}
]
},
"PBXLicense": {
    "EnableRemoteServer": true,
    "EnableProxyViaPrimaryIPOLine": false,
    "LicenseServerIPAddress": "127.0.0.1",
    "LicenseServerURL": "",
    "LicenseServerURLSecondary": "",
    "LicenseServerURN": "",
    "LicenseServerPort": 52233,
    "LicenseServerID": "000C299B41C5-WS-Nilesh",
    "LicenseMode": 4,
    "LicenseGracePeriod": 30,
    "ReservedLicenses": {
        "VMProRecordingsAdmins": 0,
        "WaveUsers": 0,
        "VMProTtsPro": 0,
        "VMProTtsWindows": 0,
        "VMProTts3rdParty": 0,
        "CtiLinkPro": 0,
        "EmbeddedVMPorts": 0,
        "VMProPorts": 2,
        "SipSessions": 0,
        "ACCS": 0,
        "BranchSystem": 0,
        "SMTrunks": 0,
        "UMSWebServices": 0,
        "MacSoftphones": 0,
        "ThirdPartyRecorder": 0,
        "ServerEdition": 1,
        "AvayaIPEndpoints": 0,
        "ThirdPartyIPEndpoints": 0,
        "Receptionist": 0,
        "BasicUser": 0,
        "OfficeWorker": 0,
        "Teleworker": 0,
        "MobileUser": 0,
        "PowerUser": 0,
        "AvayaSoftphone": 0,
        "WebCollaboration": 0,
        "UniversalPRIAdditionalchannels": 0,
        "IPSecTunneling": 0,
        "CentralizedEndpoints": 0,
        "EssentialEdition": 0,
        "PreferredEdition": 0
    },
    "LicensedVersion": "",
    "PLDSHostId": 387859159143,
    "PLDSFileStatus": 0,
    "SelectLicensingStatus": -1,
    "GraceDongleSerialNumber": "",
    "DongleSerialGraceDays": 0,
    "GracePLDSHostId": "",
    "LicensingBasicUser": true
}

```




2.8.2.2 Systems (PUT)

Method: PUT

Description: REST service which modifies a system in a solution.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/systems>

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded

Input Body:

Content body of PUT request must be of the following format:

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
    <ws_object>
        <System>
            .....
        </System>
    </ws_object>
</data>
```

XSDs: DATA, WS_OBJECT, SYSTEM

Input Parameters:

ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, update applies to the Primary.

Sample code:

Refer section [Sample code for PUT](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/systems>

Output Parameters:

- An XML document indicating the success or failure of the update request.

The response status field indicates the status, a value of 1 indicates success, 0 is a failure.

XSDs: RESPONSE, DATA, WS_OBJECT, SYSTEM, ERROR

Example 1: Update License Server information on a system - success

```
PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/systems
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=A1D589D6C7F37C31648AB2F26F8EAE96
X-User-Agent: Avaya-SDKUser
Content-Length: 268
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <System>
      <PBXLicense>
        <EnableRemoteServer>true</EnableRemoteServer>
        <LicenseServerURL>https://10.240.0.5</LicenseServerURL>
        <LicenseServerID>000433</LicenseServerID>
      </PBXLicense>
    </System>
  </ws_object>
</data>
```

JSON Request

```
{
  "ws_object": {
    "System": {
      "PBXLicense": {
        "EnableRemoteServer": "true",
        "LicenseServerURL": "https://10.240.0.5",
        "LicenseServerID": "000433"
      }
    }
  }
}
```

Result: Indicates success in updating the license server information.

XML Response

```
<response status="1">
  <data>
    <ws_object>
      <ModifiedEntities/>
    </ws_object>
    <ws_object>
      <reboot type="normal"/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
          "ModifiedEntities": ""
        },
        {
          "reboot": {
            "@type": "normal"
          }
        }
      ]
    }
  }
}
```

2.8.3 Licenses

2.8.3.1 Licenses (GET)

Description: REST service which queries/lists the licenses on a solution or node in a solution.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/licenses>

Method: GET

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- **ipaddress** (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the

entire solution.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/licenses>

Output Parameters:

An XML document containing information on each license in the solution or solution node including (but not limited to):

- Response Status – A value of 1 indicates Success, 0 is a failure
- Total Number of License Records
- *GUID* – Global Unique Identifier which uniquely identifies the license in the system.
- LicenseKey – Name of the license
- Source – The source of the license
 - Local = 0 : Indicates the license is locally installed (usually ADI license)
 - WebLM = 1 : Indicates the license is from a WebLM server.
 - MappedLocal = 2 : Indicates that the license is
 - MappedRemote = 3
 - MappedVirtual = 4
- Type
- Status – The status of the license.
 - INVALID = 0
 - VALID = 1
 - EXPIRED = 2
 - TOKENEXPIRED = 3
 - UNKNOWN = 4
 - DORMANT = 5
 - OBSOLETE = 6
- Quantity – Quantity of the license
- FreeInstances – Current free instances of the license.
- ExpiryDate – Expiry Date
- Mode
 - UNKNOWN= 0
 - ADI = 1
 - PLDS = 2
 - VIRTUAL = 3
- DisplayName – License name.

XSDs: RESPONSE, DATA, WS_OBJECT, LICENSE, ERROR

Example 1: Get all licenses in the solution.

```
GET https://173.255.118.10:7070/WebManagement/ws/sdk/admin/v1/licenses HTTP/1.1
Accept-Encoding: gzip,deflate
UseXBasic: true
Cookie: JSESSIONID=13EE1F4EFB8C76E04A7EDD13F9FBA342
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 173.255.118.10:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 01 Mar 2016 13:14:30 GMT
Content-Type: application/xml
Content-Length: 3501
Date: Tue, 01 Mar 2016 13:14:28 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data totalRecords="6">
    <ws_object>
      <License GUID="CB61B380E4B711DEB53142010AF00009">
        <LicenseKey>IP Office Select Platform</LicenseKey>
        <Source>3</Source>
        <Type>116</Type>
        <Status>1</Status>
        <Quantity>1</Quantity>
        <FreeInstances>1</FreeInstances>
        <ExpiryDate>2016/08/27</ExpiryDate>
        <Mode>2</Mode>
        <DisplayName>IP Office Select</DisplayName>
        <AvailableUserPackageLicenseOfType>1</AvailableUserPackageLicenseOfType>
        <IsGrace>false</IsGrace>
        <PBXAddress>10.240.0.9</PBXAddress>
        <Last-Modified>Mon, 29 Feb 2016 16:41:31 GMT</Last-Modified>
        <Etag>2016-02-29T16:41:31:000R2669</Etag>
      </License>
    </ws_object>
    <ws_object>
      <License GUID="CB61B380E4B711DEB52F42010AF00009">
        <LicenseKey>CTI Link Pro</LicenseKey>
        <Source>3</Source>
        <Type>2</Type>
        <Status>1</Status>
        <Quantity>1</Quantity>
        <FreeInstances>1</FreeInstances>
        <ExpiryDate>2016/08/27</ExpiryDate>
        <Mode>2</Mode>
        <DisplayName>CTI Link Pro</DisplayName>
        <AvailableUserPackageLicenseOfType>1</AvailableUserPackageLicenseOfType>
        <IsGrace>false</IsGrace>
        <PBXAddress>10.240.0.9</PBXAddress>
        <Last-Modified>Mon, 29 Feb 2016 16:41:31 GMT</Last-Modified>
        <Etag>2016-02-29T16:41:31:000R5636</Etag>
      </License>
    </ws_object>
```

```

<ws_object>
  <License GUID="CB61B380E4B711DEB53042010AF00009">
    <LicenseKey>Server Edition</LicenseKey>
    <Source>3</Source>
    <Type>123</Type>
    <Status>1</Status>
    <Quantity>1</Quantity>
    <FreeInstances>1</FreeInstances>
    <ExpiryDate>2016/08/27</ExpiryDate>
    <Mode>2</Mode>
    <DisplayName>Server Edition R10</DisplayName>
    <AvailableUserPackageLicenseOfType>1</AvailableUserPackageLicenseOfType>
    <IsGrace>false</IsGrace>
    <PBXAddress>10.240.0.9</PBXAddress>
    <Last-Modified>Mon, 29 Feb 2016 16:41:31 GMT</Last-Modified>
    <Etag>2016-02-29T16:41:31:000R1208</Etag>
  </License>
</ws_object>
<ws_object>
  <License GUID="CB61B380E4B711DEB52D42010AF00009">
    <LicenseKey>VMPro Recordings Administrators</LicenseKey>
    <Source>3</Source>
    <Type>29</Type>
    <Status>1</Status>
    <Quantity>1</Quantity>
    <FreeInstances>1</FreeInstances>
    <ExpiryDate>2016/08/27</ExpiryDate>
    <Mode>2</Mode>
    <DisplayName>VMPro Recordings Administrators</DisplayName>
    <AvailableUserPackageLicenseOfType>1</AvailableUserPackageLicenseOfType>
    <IsGrace>false</IsGrace>
    <PBXAddress>10.240.0.9</PBXAddress>
    <Last-Modified>Mon, 29 Feb 2016 16:41:31 GMT</Last-Modified>
    <Etag>2016-02-29T16:41:31:000R4503</Etag>
  </License>
</ws_object>
<ws_object>
  <License GUID="CB61B380E4B711DEB52C42010AF00009">
    <LicenseKey>Additional Voicemail Pro Ports</LicenseKey>
    <Source>3</Source>
    <Type>18</Type>
    <Status>1</Status>
    <Quantity>25</Quantity>
    <FreeInstances>25</FreeInstances>
    <ExpiryDate>2016/08/27</ExpiryDate>
    <Mode>2</Mode>
    <DisplayName>Additional Voicemail Pro (ports)</DisplayName>
    <AvailableUserPackageLicenseOfType>25</AvailableUserPackageLicenseOfType>
    <IsGrace>false</IsGrace>
    <PBXAddress>10.240.0.9</PBXAddress>
    <Last-Modified>Mon, 29 Feb 2016 16:41:31 GMT</Last-Modified>
    <Etag>2016-02-29T16:41:31:000R5801</Etag>
  </License>
</ws_object>
<ws_object>
  <License GUID="CB61B380E4B711DEB52E42010AF00009">
    <LicenseKey>SIP Trunk Channels</LicenseKey>
    <Source>3</Source>
    <Type>45</Type>
    <Status>1</Status>
    <Quantity>27</Quantity>
    <FreeInstances>27</FreeInstances>
    <ExpiryDate>2016/08/27</ExpiryDate>
    <Mode>2</Mode>
    <DisplayName>SIP Trunk Channels</DisplayName>
    <AvailableUserPackageLicenseOfType>27</AvailableUserPackageLicenseOfType>
    <IsGrace>false</IsGrace>
    <PBXAddress>10.240.0.9</PBXAddress>
    <Last-Modified>Mon, 29 Feb 2016 16:41:31 GMT</Last-Modified>
    <Etag>2016-02-29T16:41:31:000R7744</Etag>
  </License>
</ws_object>

```

```

    </License>
  </ws_object>
</data>
</response>

```

JSON Response

```

{
  "response": {
    "@status": "1",
    "data": {
      "@totalRecords": "6",
      "ws_object": [
        {
          "License": {
            "@GUID": "CB61B380E4B711DEB53142010AF00009",
            "LicenseKey": "IP Office Select Platform",
            "Source": "3",
            "Type": "116",
            "Status": "1",
            "Quantity": "1",
            "FreeInstances": "1",
            "ExpiryDate": "2016/08/27",
            "Mode": "2",
            "DisplayName": "IP Office Select",
            "AvailableUserPackageLicenseOfType": "1",
            "IsGrace": "false",
            "PBXAddress": "10.240.0.9",
          }
        },
        {
          "License": {
            "@GUID": "CB61B380E4B711DEB52F42010AF00009",
            "LicenseKey": "CTI Link Pro",
            "Source": "3",
            "Type": "2",
            "Status": "1",
            "Quantity": "1",
            "FreeInstances": "1",
            "ExpiryDate": "2016/08/27",
            "Mode": "2",
            "DisplayName": "CTI Link Pro",
            "AvailableUserPackageLicenseOfType": "1",
            "IsGrace": "false",
            "PBXAddress": "10.240.0.9",
          }
        },
        {
          "License": {
            "@GUID": "CB61B380E4B711DEB53042010AF00009",
            "LicenseKey": "Server Edition",
            "Source": "3",
            "Type": "123",
            "Status": "1",
            "Quantity": "1",
            "FreeInstances": "1",
            "ExpiryDate": "2016/08/27",
            "Mode": "2",
            "DisplayName": "Server Edition R10",
            "AvailableUserPackageLicenseOfType": "1",
            "IsGrace": "false",
            "PBXAddress": "10.240.0.9",
          }
        },
        {
          "License": {
            "@GUID": "CB61B380E4B711DEB52D42010AF00009",
            "LicenseKey": "VMPro Recordings Administrators",

```


Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Content-Type: none.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/users>

Output Parameters:

An XML document containing information on each user defined in the solution or solution node including (but not limited to):

- Response Status – A value of 1 indicates Success, 0 is a failure
- Total Number of User Records
- *GUID* – Global Unique Identifier which uniquely identifies the user in the system.
- Name – User name
- FullName
- Extension
- AssignedPackage – The type of user defined (i.e. Basic, Power...)
 - 1 = Basic User
 - 2 = Teleworker User
 - 3 = Mobile User
 - 4 = Power User
 - 5 = Office Worker User
 - 6 = Not Used
 - 7 = Centralized User
 - 8 = Non Licensed User

XSDs: RESPONSE, DATA, WS_OBJECT, USER, ERROR

Example 1: Get all users defined in the solution.

```
GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/users HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 08 Dec 2015 16:27:14 GMT
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Tue, 08 Dec 2015 16:27:12 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data set="75" totalRecords="3">
    <ws_object>
      <User GUID="2DF9B1002C7C11E08687000C29F56F71">
        <AssignedPackage>1</AssignedPackage>
        <BlockForwarding>false</BlockForwarding>
        <CanIntrude>false</CanIntrude>
        <CannotBeIntruded>true</CannotBeIntruded>
        <DisableForwardBusyNoAnsOnInt>false</DisableForwardBusyNoAnsOnInt>
        <DisableForwardUncondOnInt>false</DisableForwardUncondOnInt>
        <DNDEExceptions></DNDEExceptions>
        <DoNotDisturb>false</DoNotDisturb>
        <ExpansionType>0</ExpansionType>
        <Extension>854790</Extension>
        <FlareEnabled>false</FlareEnabled>
        <FollowMeNumber>123456</FollowMeNumber>
        <ForceAccountCode>false</ForceAccountCode>
        <ForwardHuntGroupCalls>false</ForwardHuntGroupCalls>
        <ForwardOnBusy>false</ForwardOnBusy>
        <ForwardOnNoAnswer>false</ForwardOnNoAnswer>
        <ForwardToVoicemail>false</ForwardToVoicemail>
        <ForwardUnconditional>false</ForwardUnconditional>
        <FullName>User FullName</FullName>
        <IdleLinePreference>true</IdleLinePreference>
        <LoginCode>123456</LoginCode>
        <MobilityFeatures>false</MobilityFeatures>
        <Name>User1</Name>
        <OutgoingCallBar>false</OutgoingCallBar>
        <OutOfHoursUserRights></OutOfHoursUserRights>
        <PhoneType>-1</PhoneType>
        <Priority>5</Priority>
        <Receptionist>false</Receptionist>
        <RemoteWorker>false</RemoteWorker>
        <SIPContact>123456</SIPContact>
        <SIPName>123456</SIPName>
        <SpecificBstType>-1</SpecificBstType>
        <TwinningType>0</TwinningType>
```

```

<UMSWebServices>false</UMSWebServices>
<UserRights></UserRights>
<UserRightsTimeProfile></UserRightsTimeProfile>
<VoicemailCode></VoicemailCode>
<VoicemailEmail></VoicemailEmail>
<VoicemailOn>true</VoicemailOn>
<WebCollaboration>false</WebCollaboration>
<XDirectory>false</XDirectory>
<DSSKeys>
  <DSSKey Key="1">
    <KeyType>0</KeyType>
    <Label></Label>
    <ActionObject>39</ActionObject>
    <Data>a</Data>
    <RingDelay>0</RingDelay>
    <ConflictInMultiEdit>false</ConflictInMultiEdit>
  </DSSKey>
  <DSSKey Key="2">
    <KeyType>0</KeyType>
    <Label></Label>
    <ActionObject>39</ActionObject>
    <Data>b</Data>
    <RingDelay>0</RingDelay>
    <ConflictInMultiEdit>false</ConflictInMultiEdit>
  </DSSKey>
  <DSSKey Key="3">
    <KeyType>0</KeyType>
    <Label></Label>
    <ActionObject>39</ActionObject>
    <Data>c</Data>
    <RingDelay>0</RingDelay>
    <ConflictInMultiEdit>false</ConflictInMultiEdit>
  </DSSKey>
</DSSKeys>
</User>
</ws_object>
<ws_object>
  <User GUID="0B67A4002C7C11E08678000C29F56F71">
    <AssignedPackage>1</AssignedPackage>
    <BlockForwarding>false</BlockForwarding>
    <CanIntrude>false</CanIntrude>
    <CannotBeIntruded>true</CannotBeIntruded>
    <DisableForwardBusyNoAnsOnInt>false</DisableForwardBusyNoAnsOnInt>
    <DisableForwardUncondOnInt>false</DisableForwardUncondOnInt>
    <DNDEExceptions></DNDEExceptions>
    <DoNotDisturb>false</DoNotDisturb>
    <ExpansionType>0</ExpansionType>
    <Extension>123456</Extension>
    <FlareEnabled>false</FlareEnabled>
    <FollowMeNumber></FollowMeNumber>
    <ForceAccountCode>false</ForceAccountCode>
    <ForwardHuntGroupCalls>false</ForwardHuntGroupCalls>
    <ForwardOnBusy>false</ForwardOnBusy>
    <ForwardOnNoAnswer>false</ForwardOnNoAnswer>
    <ForwardToVoicemail>false</ForwardToVoicemail>
    <ForwardUnconditional>false</ForwardUnconditional>
    <FullName>User FullName</FullName>
    <IdleLinePreference>true</IdleLinePreference>
    <LoginCode>123456</LoginCode>
    <MobilityFeatures>false</MobilityFeatures>
    <Name>User2</Name>
    <OutgoingCallBar>false</OutgoingCallBar>
    <OutOfHoursUserRights></OutOfHoursUserRights>
    <PhoneType>50</PhoneType>
    <Priority>5</Priority>
    <Receptionist>false</Receptionist>
    <RemoteWorker>false</RemoteWorker>
    <SIPContact>123456</SIPContact>
    <SIPName>123456</SIPName>
    <SpecificBstType>-1</SpecificBstType>
  </User>
</ws_object>

```

```

<TwinningType>0</TwinningType>
<UMSWebServices>false</UMSWebServices>
<UserRights></UserRights>
<UserRightsTimeProfile></UserRightsTimeProfile>
<VoicemailCode></VoicemailCode>
<VoicemailEmail></VoicemailEmail>
<VoicemailOn>true</VoicemailOn>
<WebCollaboration>false</WebCollaboration>
<XDirectory>false</XDirectory>
<DSSKeys>
  <DSSKey Key="1">
    <KeyType>0</KeyType>
    <Label></Label>
    <ActionObject>39</ActionObject>
    <Data>a</Data>
    <RingDelay>0</RingDelay>
    <ConflictInMultiEdit>false</ConflictInMultiEdit>
  </DSSKey>
  <DSSKey Key="2">
    <KeyType>0</KeyType>
    <Label></Label>
    <ActionObject>39</ActionObject>
    <Data>b</Data>
    <RingDelay>0</RingDelay>
    <ConflictInMultiEdit>false</ConflictInMultiEdit>
  </DSSKey>
  <DSSKey Key="3">
    <KeyType>0</KeyType>
    <Label></Label>
    <ActionObject>39</ActionObject>
    <Data>c</Data>
    <RingDelay>0</RingDelay>
    <ConflictInMultiEdit>false</ConflictInMultiEdit>
  </DSSKey>
</DSSKeys>
</User>
</ws_object>
<ws_object>
  <User GUID="679FB0002AF411E0B47F000C29F56F71">
    <AssignedPackage>8</AssignedPackage>
    <BlockForwarding>false</BlockForwarding>
    <CanIntrude>false</CanIntrude>
    <CannotBeIntruded>false</CannotBeIntruded>
    <DisableForwardBusyNoAnsOnInt>false</DisableForwardBusyNoAnsOnInt>
    <DisableForwardUncondOnInt>false</DisableForwardUncondOnInt>
    <DNDEExceptions></DNDEExceptions>
    <DoNotDisturb>false</DoNotDisturb>
    <ExpansionType>0</ExpansionType>
    <Extension></Extension>
    <FlareEnabled>false</FlareEnabled>
    <FollowMeNumber></FollowMeNumber>
    <ForceAccountCode>false</ForceAccountCode>
    <ForwardHuntGroupCalls>false</ForwardHuntGroupCalls>
    <ForwardOnBusy>false</ForwardOnBusy>
    <ForwardOnNoAnswer>false</ForwardOnNoAnswer>
    <ForwardToVoicemail>false</ForwardToVoicemail>
    <ForwardUnconditional>false</ForwardUnconditional>
    <FullName></FullName>
    <IdleLinePreference>true</IdleLinePreference>
    <LoginCode></LoginCode>
    <MobilityFeatures>false</MobilityFeatures>
    <Name>NoUser</Name>
    <OutgoingCallBar>true</OutgoingCallBar>
    <OutOfHoursUserRights></OutOfHoursUserRights>
    <PhoneType>-1</PhoneType>
    <Priority>5</Priority>
    <Receptionist>false</Receptionist>
    <RemoteWorker>false</RemoteWorker>
    <SIPContact>SIPDefault</SIPContact>
    <SIPName>SIPDefault</SIPName>
  </User>
</ws_object>

```

```

<SpecificBstType>-1</SpecificBstType>
<TwinningType>0</TwinningType>
<UMSWebServices>>false</UMSWebServices>
<UserRights></UserRights>
<UserRightsTimeProfile></UserRightsTimeProfile>
<VoicemailCode></VoicemailCode>
<VoicemailEmail></VoicemailEmail>
<VoicemailOn>true</VoicemailOn>
<WebCollaboration>>false</WebCollaboration>
<XDirectory>>false</XDirectory>
<DSSKeys>
  <DSSKey Key="1">
    <KeyType>1</KeyType>
    <Label></Label>
    <ActionObject>46</ActionObject>
    <Data></Data>
    <RingDelay>0</RingDelay>
    <ConflictInMultiEdit>>false</ConflictInMultiEdit>
  </DSSKey>
</DSSKeys>
</User>
</ws_object>
</data>
</response>

```

JSON Response

```

{"response": {
  "@status": "1",
  "data": {
    "@set": "75",
    "@totalRecords": "3",
    "ws_object": [
      { "User": {
        "@GUID": "2DF9B1002C7C11E08687000C29F56F71",
        "AssignedPackage": 1,
        "BlockForwarding": false,
        "CanIntrude": false,
        "CannotBeIntruded": true,
        "DisableForwardBusyNoAnsOnInt": false,
        "DisableForwardUncondOnInt": false,
        "DNExceptions": "",
        "DoNotDisturb": false,
        "ExpansionType": 0,
        "Extension": 854790,
        "FlareEnabled": false,
        "FollowMeNumber": 123456,
        "ForceAccountCode": false,
        "ForwardHuntGroupCalls": false,
        "ForwardOnBusy": false,
        "ForwardOnNoAnswer": false,
        "ForwardToVoicemail": false,
        "ForwardUnconditional": false,
        "FullName": "User FullName",
        "IdleLinePreference": true,
        "LoginCode": 123456,
        "MobilityFeatures": false,
        "Name": "User1",
        "OutgoingCallBar": false,
        "OutOfHoursUserRights": "",
        "PhoneType": -1,
        "Priority": 5,
        "Receptionist": false,
        "RemoteWorker": false,
        "SIPContact": 123456,
        "SIPName": 123456,
        "SpecificBstType": -1,
        "TwinningType": 0,
        "UMSWebServices": false,

```

```

    "UserRights": "",
    "UserRightsTimeProfile": "",
    "VoicemailCode": "",
    "VoicemailEmail": "",
    "VoicemailOn": true,
    "WebCollaboration": false,
    "XDirectory": false,
    "DSSKeys": {"DSSKey": [
        {
            "@Key": "1",
            "KeyType": 0,
            "Label": "",
            "ActionObject": 39,
            "Data": "a=",
            "RingDelay": 0,
            "ConflictInMultiEdit": false
        },
        {
            "@Key": "2",
            "KeyType": 0,
            "Label": "",
            "ActionObject": 39,
            "Data": "b=",
            "RingDelay": 0,
            "ConflictInMultiEdit": false
        },
        {
            "@Key": "3",
            "KeyType": 0,
            "Label": "",
            "ActionObject": 39,
            "Data": "c=",
            "RingDelay": 0,
            "ConflictInMultiEdit": false
        }
    ]}
  }},
  {"User": {
    "@GUID": "0B67A4002C7C11E08678000C29F56F71",
    "AssignedPackage": 1,
    "BlockForwarding": false,
    "CanIntrude": false,
    "CannotBeIntruded": true,
    "DisableForwardBusyNoAnsOnInt": false,
    "DisableForwardUncondOnInt": false,
    "DNDEExceptions": "",
    "DoNotDisturb": false,
    "ExpansionType": 0,
    "Extension": 123456,
    "FlareEnabled": false,
    "FollowMeNumber": "",
    "ForceAccountCode": false,
    "ForwardHuntGroupCalls": false,
    "ForwardOnBusy": false,
    "ForwardOnNoAnswer": false,
    "ForwardToVoicemail": false,
    "ForwardUnconditional": false,
    "FullName": "User FullName",
    "IdleLinePreference": true,
    "LoginCode": 123456,
    "MobilityFeatures": false,
    "Name": "User2",
    "OutgoingCallBar": false,
    "OutOfHoursUserRights": "",
    "PhoneType": 50,
    "Priority": 5,
    "Receptionist": false,
    "RemoteWorker": false,
    "SIPContact": 123456,
    "SIPName": 123456,

```

```

"SpecificBstType": -1,
"TwiningType": 0,
"UMSWebServices": false,
"UserRights": "",
"UserRightsTimeProfile": "",
"VoicemailCode": "",
"VoicemailEmail": "",
"VoicemailOn": true,
"WebCollaboration": false,
"XDirectory": false,
"DSSKeys": {"DSSKey": [
  {
    "@Key": "1",
    "KeyType": 0,
    "Label": "",
    "ActionObject": 39,
    "Data": "a=",
    "RingDelay": 0,
    "ConflictInMultiEdit": false
  },
  {
    "@Key": "2",
    "KeyType": 0,
    "Label": "",
    "ActionObject": 39,
    "Data": "b=",
    "RingDelay": 0,
    "ConflictInMultiEdit": false
  },
  {
    "@Key": "3",
    "KeyType": 0,
    "Label": "",
    "ActionObject": 39,
    "Data": "c=",
    "RingDelay": 0,
    "ConflictInMultiEdit": false
  }
]}
}},
{"User": {
  "@GUID": "679FB0002AF411E0B47F000C29F56F71",
  "AssignedPackage": 8,
  "BlockForwarding": false,
  "CanIntrude": false,
  "CannotBeIntruded": false,
  "DisableForwardBusyNoAnsOnInt": false,
  "DisableForwardUncondOnInt": false,
  "DNDEExceptions": "",
  "DoNotDisturb": false,
  "ExpansionType": 0,
  "Extension": "",
  "FlareEnabled": false,
  "FollowMeNumber": "",
  "ForceAccountCode": false,
  "ForwardHuntGroupCalls": false,
  "ForwardOnBusy": false,
  "ForwardOnNoAnswer": false,
  "ForwardToVoicemail": false,
  "ForwardUnconditional": false,
  "FullName": "",
  "IdleLinePreference": true,
  "LoginCode": "",
  "MobilityFeatures": false,
  "Name": "NoUser",
  "OutgoingCallBar": true,
  "OutOfHoursUserRights": "",
  "PhoneType": -1,
  "Priority": 5,
  "Receptionist": false,

```

Description: REST service which adds a user to a solution node.

Method: POST

Headers:

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Body:

Content body of POST request must be of the following format:

Management API IP Office™ Platform Issue 3.00


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data set="75" totalRecords="1">
    <ws_object>
      <User>
        <AssignedPackage> </AssignedPackage>
        <CanIntrude> </CanIntrude>
        <CannotBeIntruded> </CannotBeIntruded>
        <DNDEExceptions/>
        <DoNotDisturb> </DoNotDisturb>
        <ExpansionType> </ExpansionType>
        <Extension/>
        <FlareEnabled> </FlareEnabled>
        <ForceAccountCode> </ForceAccountCode>
        <FullName> </FullName>
        <IdleLinePreference> </IdleLinePreference>
        <LoginCode> </LoginCode>
        <MobilityFeatures> </MobilityFeatures>
        <Name> </Name>
        <OutgoingCallBar> </OutgoingCallBar>
        <OutOfHoursUserRights> </OutOfHoursUserRights>
        <Password> </Password>
        <PhoneType> </PhoneType>
        <Priority> </Priority>
        <Receptionist> </Receptionist>
        <RemoteWorker> </RemoteWorker>
        <SIPContact> </SIPContact>
        <SIPName> </SIPName>
        <SpecificBstType> </SpecificBstType>
        <TwinningType> </TwinningType>
        <UMSWebServices> </UMSWebServices>
        <UserRights> </UserRights>
        <UserRightsTimeProfile></UserRightsTimeProfile>
        <VoicemailCode> </VoicemailCode>
        <VoicemailEmail/> </VoicemailEmail>
        <VoicemailOn> </VoicemailOn>
        <WebCollaboration> </WebCollaboration>
        <XDirectory> </XDirectory>
      </User>
    </ws_object>
  </data>
</response>
```

JSON Request Template

```
{
  "response": {
    "@status": "1",
    "data": {
      "@set": "75",
      "@totalRecords": "1",
      "ws_object": {
        "User": {
          "AssignedPackage": " ",
          "CanIntrude": " ",
          "CannotBeIntruded": " ",
          "DNDEExceptions": " ",
          "DoNotDisturb": " ",
          "ExpansionType": " ",
          "Extension": " ",
          "FlareEnabled": " ",
          "ForceAccountCode": " ",
          "FullName": " ",
          "IdleLinePreference": " ",
          "LoginCode": " ",
          "MobilityFeatures": " ",
          "Name": " ",
          "OutgoingCallBar": " ",
```

```

        "OutOfHoursUserRights": " ",
        "Password": " ",
        "PhoneType": " ",
        "Priority": " ",
        "Receptionist": " ",
        "RemoteWorker": " ",
        "SIPContact": "",
        "SIPName": " ",
        "SpecificBstType": " ",
        "TwinningType": " ",
        "UMSWebServices": " ",
        "UserRights": " ",
        "VoicemailCode": " ",
        "VoicemailEmail": " ",
        "VoicemailOn": " ",
        "WebCollaboration": " ",
        "XDirectory": " "
    }
}
}
}
}

```

NOTE: <ws_object> block can be repeated to add multiple users in a single request.

XSDs: DATA, WS_OBJECT, USER

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.

Sample code:

Refer section [Sample code for POST](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/users>

Output Parameters:

An XML document indicating the success or failure of the Add request.

XSDs: RESPONSE, DATA, WS_OBJECT, USER, ERROR

Example 1: Add 1 new user to a system - failure

```

POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/users
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=A1D589D6C7F37C31648AB2F26F8EAE96
X-User-Agent: Avaya-SDKUser
Content-Length: 874
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

```

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <User>
      <Name>ethel</Name>
      <Password>195656</Password>
      <FullName>ethel E129</FullName>
      <Extension>309</Extension>
      <VoicemailEmail/>
      <XDirectory>false</XDirectory>
      <VoicemailOn>true</VoicemailOn>
      <VoicemailCode>1956</VoicemailCode>
      <Receptionist>false</Receptionist>
      <AssignedPackage>4</AssignedPackage>
      <RemoteWorker>true</RemoteWorker>
      <TwinningType>0</TwinningType>
      <MobilityFeatures>false</MobilityFeatures>
      <UMSWebServices>true</UMSWebServices>
      <FlareEnabled>true</FlareEnabled>
      <LoginCode>195656</LoginCode>
      <SIPName/>
      <SIPContact/>
      <WebCollaboration>false</WebCollaboration>
    </User>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "User": {
        "Name": "ethel",
        "Password": "195656",
        "FullName": "ethel E129",
        "Extension": "309",
        "XDirectory": "false",
        "VoicemailOn": "true",
        "VoicemailCode": "1956",
        "Receptionist": "false",
        "AssignedPackage": "4",
        "RemoteWorker": "true",
        "TwinningType": "0",
        "MobilityFeatures": "false",
        "UMSWebServices": "true",
        "FlareEnabled": "true",
        "LoginCode": "195656",
        "WebCollaboration": "false"
      }
    }
  }
}
```

Result: Indicates 2 errors with the add request. One error is that the user name already exists and the second is to indicate that the extension provided also conflicts with another user's extension.

XML Response

```
<?xml version='1.0' encoding='UTF-8'?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <USER>
        <error>
          <error_code>9020</error_code>
          <error_desc>"A user or huntgroup with the specified Name Already
Exist."</error_desc>
          <cause>0</cause>
          <cause_data>ethel</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
        <error>
          <error_code>18011</error_code>
          <error_desc>"Extension conflicts with another user"</error_desc>
          <cause>0</cause>
          <cause_data>ethel</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </USER>
    </ws_object>
    <ws_object error_node="true">
      <REBOOT>
        <error>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </REBOOT>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": [
        {
          "@error_node": "true",
          "USER": {
            "error": [
              {
                "error_code": "9020",
                "error_desc": "\"A user or huntgroup with the specified Name Already
Exist.\"\"",
                "cause": "0",
                "cause_data": "ethel",
                "errorLocation": "10.240.0.5"
              },
              {
                "error_code": "18011",
                "error_desc": "\"Extension conflicts with another user\"",
                "cause": "0",
                "cause_data": "ethel",
                "errorLocation": "10.240.0.5"
              }
            ]
          }
        },
        {
          "@error_node": "true",
          "REBOOT": {
            "error": { "errorLocation": "10.240.0.5" }
          }
        }
      ]
    }
  }
}
```

```
}
  }
}
}
```

Example 2: Add 1 new user to a system – failure – session timeout

POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/users
 Accept-Encoding: gzip,deflate
 Content-Type: application/xml
 X-User-Client: Avaya-WebAdmin
 Cookie: JSESSIONID=A1D589D6C7F37C31648AB2F26F8EAE96
 X-User-Agent: Avaya-SDKUser
 Content-Length: 874
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <User>
      <Name>ethel</Name>
      <Password>195656</Password>
      <FullName>ethel E129</FullName>
      <Extension>309</Extension>
      <VoicemailEmail/>
      <XDirectory>false</XDirectory>
      <VoicemailOn>true</VoicemailOn>
      <VoicemailCode>1956</VoicemailCode>
      <Receptionist>false</Receptionist>
      <AssignedPackage>4</AssignedPackage>
      <RemoteWorker>true</RemoteWorker>
      <TwinningType>0</TwinningType>
      <MobilityFeatures>false</MobilityFeatures>
      <UMSWebServices>true</UMSWebServices>
      <FlareEnabled>true</FlareEnabled>
      <LoginCode>195656</LoginCode>
      <SIPName/>
      <SIPContact/>
      <WebCollaboration>false</WebCollaboration>
    </User>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "User": {
        "Name": "ethel",
        "Password": "195656",
        "FullName": "ethel E129",
        "Extension": "309",
        "XDirectory": "false",
        "VoicemailOn": "true",
        "VoicemailCode": "1956",
        "Receptionist": "false",
        "AssignedPackage": "4",
        "RemoteWorker": "true",
```

```

    "TwinningType": "0",
    "MobilityFeatures": "false",
    "UMSWebServices": "true",
    "FlareEnabled": "true",
    "LoginCode": "195656",
    "WebCollaboration": "false"
  }
}
}

```

Result: Indicates an error because the session has timed out. A new session would need to be created by authenticating again.

XML Response

```

<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>461:SessionTimeout</error_code>
          <error_desc>130.211.136.50</error_desc>
          <errorLocation>130.211.136.50</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>

```

JSON Response

```

{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
            "error_code": "461:SessionTimeout",
            "error_desc": "130.211.136.50",
            "errorLocation": "130.211.136.50"
          }
        }
      }
    }
  }
}

```

Example 3: Add 1 new user to a system – success

```

POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/users
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=A1D589D6C7F37C31648AB2F26F8EAE96
X-User-Agent: Avaya-SDKUser
Content-Length: 879
Host: 130.211.136.50:7070

```

Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <User>
      <Name>ethelmania</Name>
      <Password>195656</Password>
      <FullName>ethel E129</FullName>
      <Extension>399</Extension>
      <VoicemailEmail/>
      <XDirectory>false</XDirectory>
      <VoicemailOn>true</VoicemailOn>
      <VoicemailCode>1956</VoicemailCode>
      <Receptionist>false</Receptionist>
      <AssignedPackage>4</AssignedPackage>
      <RemoteWorker>true</RemoteWorker>
      <TwinningType>0</TwinningType>
      <MobilityFeatures>false</MobilityFeatures>
      <UMSWebServices>true</UMSWebServices>
      <FlareEnabled>true</FlareEnabled>
      <LoginCode>195656</LoginCode>
      <SIPName/>
      <SIPContact/>
      <WebCollaboration>false</WebCollaboration>
    </User>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "User": {
        "Name": "ethelmania",
        "Password": "195656",
        "FullName": "ethel E129",
        "Extension": "399",
        "XDirectory": "false",
        "VoicemailOn": "true",
        "VoicemailCode": "1956",
        "Receptionist": "false",
        "AssignedPackage": "4",
        "RemoteWorker": "true",
        "TwinningType": "0",
        "MobilityFeatures": "false",
        "UMSWebServices": "true",
        "FlareEnabled": "true",
        "LoginCode": "195656",
        "WebCollaboration": "false"
      }
    }
  }
}
```

Result: Indicates success and that a new user has been successfully created. The GUID for the new user is returned in the response.

XML Response

```
<response status="1">
  <data>
    <ws_object>
      <User GUID="70BD92809BC511DEBA7242010AF00005">
        <Name>ethelmania</Name>
      </User>
    </ws_object>
    <ws_object>
      <reboot type="normal"/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
          "User": {
            "@GUID": "70BD92809BC511DEBA7242010AF00005",
            "Name": "ethelmania"
          }
        },
        {
          "reboot": { "@type": "normal" }
        }
      ]
    }
  }
}
```

2.8.4.3 Users (PUT)

Description: REST service which updates an existing user on a system.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/users>

Method: PUT

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Content-Length: <length in bytes of the xml body>

Input Body:

Content body of PUT request must be of the following format (note that the GUID of the user to update must be included as an attribute of the User):

XML Request Template

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data set="75" totalRecords="1">
    <ws_object>
      <User GUID="<guid of the user">
        <AssignedPackage> </AssignedPackage>
        <CanIntrude> </CanIntrude>
        <CannotBeIntruded> </CannotBeIntruded>
        <DNDEExceptions/>
        <DoNotDisturb> </DoNotDisturb>
        <ExpansionType> </ExpansionType>
        <Extension/>
        <FlareEnabled> </FlareEnabled>
        <ForceAccountCode> </ForceAccountCode>
        <FullName> </FullName>
        <IdleLinePreference> </IdleLinePreference>
        <LoginCode> </LoginCode>
        <MobilityFeatures> </MobilityFeatures>
        <Name> </Name>
        <OutgoingCallBar> </OutgoingCallBar>
        <OutOfHoursUserRights> </OutOfHoursUserRights>
        <Password> </Password>
        <PhoneType> </PhoneType>
        <Priority> </Priority>
        <Receptionist> </Receptionist>
        <RemoteWorker> </RemoteWorker>
        <SIPContact> </SIPContact>
        <SIPName> </SIPName>
        <SpecificBstType> </SpecificBstType>
        <TwinningType> </TwinningType>
        <UMSWebServices> </UMSWebServices>
        <UserRights> </UserRights>
        <UserRightsTimeProfile></UserRightsTimeProfile>
        <VoicemailCode> </VoicemailCode>
        <VoicemailEmail> </VoicemailEmail>
        <VoicemailOn> </VoicemailOn>
        <WebCollaboration> </WebCollaboration>
        <XDirectory> </XDirectory>
      </User>
    </ws_object>
  </data>
</response>
```

JSON Request Template

```
{
  "response": {
    "@status": "1",
    "data": {
      "@set": "75",
      "@totalRecords": "1",
      "ws_object": {
        "User": {
          "@GUID": "guidOfTheUser",
          "AssignedPackage": " ",
          "CanIntrude": " ",
          "CannotBeIntruded": " ",
          "DoNotDisturb": " ",
          "ExpansionType": " ",
          "FlareEnabled": " ",
```

```

    "ForceAccountCode": "    ",
    "FullName": "        ",
    "IdleLinePreference": "    ",
    "LoginCode": "        ",
    "MobilityFeatures": "    ",
    "Name": "            ",
    "OutgoingCallBar": "    ",
    "OutOfHoursUserRights": "    ",
    "Password": "        ",
    "PhoneType": "        ",
    "Priority": "        ",
    "Receptionist": "    ",
    "RemoteWorker": "    ",
    "SIPContact": "        ",
    "SIPName": "        ",
    "SpecificBstType": "    ",
    "TwinningType": "        ",
    "UMSWebServices": "    ",
    "UserRights": "        ",
    "VoicemailCode": "        ",
    "VoicemailEmail": "    ",
    "VoicemailOn": "        ",
    "WebCollaboration": "    ",
    "XDirectory": "        "
  }
}
}
}

```

NOTE: <ws_object> block can be repeated to update multiple users in a single request.

XSDs: DATA, WS_OBJECT, USER

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.

Sample code:

Refer section [Sample code for PUT](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/users>

Output Parameters:

An XML document indicating the success or failure of the update request.

XSDs: RESPONSE, DATA, WS_OBJECT, USER, ERROR

Example 1: Update a user in the system – failure – No GUID

```

PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/users HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=6FE5AC38C87FC9A8B91F8E5EBA09B00E

```

X-User-Agent: Avaya-SDKUser
 Content-Length: 172
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <User>
      <FullName>ethel E130</FullName>
    </User>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "User": { "FullName": "ethel E130" }
    }
  }
}
```

Result: Indicates an error with the update request. The error indicates that no GUID was supplied to identify the user.

XML Response

```
<response status="0">
  <data>
    <ws_object error_node="true">
      <USER>
        <error>
          <error_code>15</error_code>
          <error_desc>"GUID missing in body"</error_desc>
          <cause>0</cause>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </USER>
    </ws_object>
    <ws_object error_node="true">
      <REBOOT>
        <error>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </REBOOT>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": [
        {
          "@error_node": "true",
          "USER": {
            "error": {
              "error_code": "15",
              "error_desc": "\"GUID missing in body\"",

```

```

        "cause": "0",
        "errorLocation": "10.240.0.5"
    }
},
{
    "@error_node": "true",
    "REBOOT": {
        "error": { "errorLocation": "10.240.0.5" }
    }
}
]
}
}
}

```

Example 2: Update an existing user's Full Name.

PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/users HTTP/1.1
 Accept-Encoding: gzip,deflate
 Content-Type: application/xml
 X-User-Client: Avaya-WebAdmin
 Cookie: JSESSIONID=6FE5AC38C87FC9A8B91F8E5EBA09B00E
 X-User-Agent: Avaya-SDKUser
 Content-Length: 212
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```

<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <User GUID="70BD92809BC511DEBA7242010AF00005">
      <FullName>ethel E129</FullName>
    </User>
  </ws_object>
</data>

```

JSON Request

```

{
  "data": {
    "ws_object": {
      "User": {
        "-GUID": "70BD92809BC511DEBA7242010AF00005",
        "FullName": "ethel E129"
      }
    }
  }
}

```

Result: Indicates a successful update of the user.

XML Response

```
<response status="1">
  <data>
    <ws_object>
      <reboot type="normal"/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": {
        "reboot": { "@type": "normal" }
      }
    }
  }
}
```

2.8.4.4 Users (DELETE)

Description: REST service which deletes an existing user on a system.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/users>

Method: DELETE

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- **ipaddress** (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.
- **Guid** (required) – GUID of the user to delete from the system.

Sample code:

Refer section [Sample code for DELETE](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/users>

Output Parameters:

An XML document indicating the success or failure of the delete request.

XSDs: RESPONSE, DATA, WS_OBJECT, USER, ERROR

Example 1: Delete a user in the system – failure – No GUID

```
DELETE https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/users HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=6FE5AC38C87FC9A8B91F8E5EBA09B00E
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result: Indicates an error with the delete request. The reason was no GUID was supplied.

XML Response

```
<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>400:BadRequest</error_code>
          <error_desc>10.240.0.5</error_desc>
          <cause/>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
            "error_code": "400:BadRequest",
            "error_desc": "10.240.0.5",
            "errorLocation": "10.240.0.5"
          }
        }
      }
    }
  }
}
```

Example 2: Delete a user in the system – failure – GUID no found

```
DELETE https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/users
HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=6FE5AC38C87FC9A8B91F8E5EBA09B00E
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result: Indicates an error with the delete request. The reason was no GUID was supplied.

XML Response

```
<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>404:NotFound</error_code>
          <error_desc>10.240.0.5</error_desc>
          <cause/>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
            "error_code": "404:NotFound",
            "error_desc": "10.240.0.5",
            "errorLocation": "10.240.0.5"
          }
        }
      }
    }
  }
}
```

Example 3: Delete an existing user - success

```
DELETE https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/users?guid=<GUID> of
the user to be deleted>
HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=6FE5AC38C87FC9A8B91F8E5EBA09B00E
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result: Indicates a successful deletion of the user.

XML Response

```
<response status="1">
  <data>
    <ws_object>
      <ModifiedEntities/>
    </ws_object>
    <ws_object>
      <reboot type="normal"/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
        },
        {
          "reboot": { "@type": "normal" }
        }
      ]
    }
  }
}
```

2.8.5 Extensions

2.8.5.1 Extensions (GET)

Description: REST service which queries/lists extensions on a solution or node in a solution.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/extensions>

Method: GET

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- `ipaddress` (optional) – LAN 1 IP Address of the Solution Node on which to

apply the command. If not present in the request, query applies to the entire solution.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/extensions>

Output Parameters:

An XML document containing information on each extension defined in the solution or solution node including (but not limited to):

- Response Status – A value of 1 indicates Success, 0 is a failure
- Total Number of Extension Records
- *GUID* – Global Unique Identifier which uniquely identifies the extension in the system.
- Extension – extension number
- TypeInfo – The type of the extension which could translate to an H.323, SIP, Digital, Analog ... device.

XSDs: RESPONSE, DATA, WS_OBJECT, EXTENSION, ERROR

Example 1: Get all extensions defined in the solution – Error Session Timeout

```
GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/extensions HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=6FE5AC38C87FC9A8B91F8E5EBA09B00E
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

XML Response

```
<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>461:SessionTimeout</error_code>
          <error_desc>130.211.136.50</error_desc>
          <errorLocation>130.211.136.50</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
            "error_code": "461:SessionTimeout",
            "error_desc": "130.211.136.50",
            "errorLocation": "130.211.136.50"
          }
        }
      }
    }
  }
}
```

Example 2: Get all extensions defined in the solution – Success

GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/extensions HTTP/1.1
 Accept-Encoding: gzip,deflate
 Cookie: JSESSIONID=6FBDC5BED65E340D50DFE09A703F8973
 X-User-Client: Avaya-WebAdmin
 X-User-Agent: Avaya-SDKUser
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Fri, 27 Nov 2015 13:19:19 GMT
 Content-Type: application/xml
 Content-Length: 6419
 Date: Fri, 27 Nov 2015 13:19:17 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data set="75" totalRecords="26">
    <ws_object>
      <Extension GUID="31A7D5009AC511DEAE0C42010AF00005">
        <Id>11218</Id>
        <Extension>306</Extension>
        <TypeInfo>1</TypeInfo>
        <CallerDisplayType>1</CallerDisplayType>
        <Module>0</Module>
        <Port>0</Port>
        <Location>0</Location>
      </Extension>
    </ws_object>
    <ws_object>
      <Extension GUID="2CE27B009AC511DEAE0C42010AF00005">
        <Id>11216</Id>
        <Extension>314</Extension>
        <TypeInfo>4</TypeInfo>
        <CallerDisplayType>1</CallerDisplayType>
        <Module>0</Module>
```

```

    <Port>0</Port>
    <Location>0</Location>
  </Extension>
</ws_object>
<ws_object>
  <Extension GUID="1DFC21809AC511DEAE0842010AF00005">
    <Id>11207</Id>
    <Extension>312</Extension>
    <TypeInfo>3</TypeInfo>
    <CallerDisplayType>1</CallerDisplayType>
    <Module>0</Module>
    <Port>0</Port>
    <Location>0</Location>
  </Extension>
</ws_object>
</data>
</response>

```

JSON Response

```

{
  "response": {
    "@status": "1",
    "data": {
      "@set": "75",
      "@totalRecords": "26",
      "ws_object": [
        {
          "Extension": {
            "@GUID": "31A7D5009AC511DEAE0C42010AF00005",
            "Id": "11218",
            "Extension": "306",
            "TypeInfo": "1",
            "CallerDisplayType": "1",
            "Module": "0",
            "Port": "0",
            "Location": "0"
          }
        },
        {
          "Extension": {
            "@GUID": "2CE27B009AC511DEAE0C42010AF00005",
            "Id": "11216",
            "Extension": "314",
            "TypeInfo": "4",
            "CallerDisplayType": "1",
            "Module": "0",
            "Port": "0",
            "Location": "0"
          }
        },
        {
          "Extension": {
            "@GUID": "1DFC21809AC511DEAE0842010AF00005",
            "Id": "11207",
            "Extension": "312",
            "TypeInfo": "3",
            "CallerDisplayType": "1",
            "Module": "0",
            "Port": "0",
            "Location": "0"
          }
        }
      ]
    }
  }
}

```

2.8.5.2 Extensions (POST)

Description: REST service which adds an extension to a solution node.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/extensions

Method: POST

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Content-Length: <length in bytes of the xml body>

Input Body:

Content body of POST request must be of the following format:

XML Request

```
<?xml version="1.0" encoding="UTF-8" ?>
<data>
  <ws_object>
    <Extension>
      <Extension>      </Extension>
      <TypeInfo>      </TypeInfo>
    </Extension>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "Extension": {
        "Extension": " ",
        "TypeInfo": " "
      }
    }
  }
}
```

NOTE: <ws_object> block can be repeated to add multiple extensions in a single request.

XSDs: DATA, WS_OBJECT, EXTENSION

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the

Primary.

Sample code:

Refer section [Sample code for POST](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/systems>

Output Parameters:

.An XML document indicating the success or failure of the Add request.

XSDs: RESPONSE, DATA, WS_OBJECT, EXTENSION, ERROR

Example 1: Add 1 new extension to a system - success

```
POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/extensions HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Cookie: JSESSIONID=9413A0345BE1E59DC0EA0DB80467FAE9
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Content-Length: 154
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

XML Request

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <Extension>
      <Extension>406</Extension>
      <TypeInfo>1</TypeInfo>
    </Extension>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "Extension": {
        "Extension": "406",
        "TypeInfo": "1"
      }
    }
  }
}
```

Result: Indicates success and returns the GUID of the newly created extension along with an Id.

XML Response

```
<?xml version='1.0' encoding='UTF-8' ?>
<response status="1">
```

```
<data>
  <ws_object>
    <Extension GUID="AAD783809C5711DEBD3842010AF00005">
      <Id>11250</Id>
    </Extension>
  </ws_object>
  <ws_object>
    <ModifiedEntities/>
  </ws_object>
  <ws_object>
    <reboot type="normal"/>
  </ws_object>
</data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
          "Extension": {
            "@GUID": "AAD783809C5711DEBD3842010AF00005",
            "Id": "11250"
          }
        },
        {
          "reboot": { "@type": "normal" }
        }
      ]
    }
  }
}
```

Example 2: Add 1 new extension to a system – failure – already exists

POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/extensions HTTP/1.1
 Accept-Encoding: gzip,deflate
 Content-Type: application/xml
 X-User-Client: Avaya-WebAdmin
 Cookie: JSESSIONID=9413A0345BE1E59DC0EA0DB80467FAE9
 X-User-Agent: Avaya-SDKUser
 Content-Length: 154
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <Extension>
      <Extension>406</Extension>
      <TypeInfo>1</TypeInfo>
    </Extension>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "Extension": {
        "Extension": "406",
        "TypeInfo": "1"
      }
    }
  }
}
```

Result: Indicates an error because the extension already exists.

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Fri, 27 Nov 2015 14:25:28 GMT
 Content-Type: application/xml
 Content-Length: 489
 Date: Fri, 27 Nov 2015 14:25:26 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <EXTENSION>
        <error>
          <error_code>9042</error_code>
          <error_desc>"Extension number "406" already exists"</error_desc>
          <cause>0</cause>
          <cause_data>0</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </EXTENSION>
    </ws_object>
    <ws_object error_node="true">
      <REBOOT>
        <error>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </REBOOT>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": [
        {
          "@error_node": "true",
          "EXTENSION": {
            "error": {
              "error_code": "9042",
              "error_desc": "\"Extension number \"406\" already exists\"",
              "cause": "0",
              "cause_data": "0",
              "errorLocation": "10.240.0.5"
            }
          }
        }
      ]
    }
  }
}
```

```

    },
    {
      "@error_node": "true",
      "REBOOT": {
        "error": { "errorLocation": "10.240.0.5" }
      }
    }
  ]
}
}
}

```

2.8.5.3 Extensions (PUT)

Description: REST service which updates an existing extension on a system.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/extensions>

Method: PUT

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Content-Length: <length in bytes of the xml body>

Input Body:

Content body of PUT request must be of the following format (note that the GUID of the extension to update must be included as an attribute of the Extension):

XML Request

```

<?xml version="1.0" encoding="UTF-8" ?>
<data>
  <ws_object>
    <Extension GUID="<guid of extension>">
      <Extension>      </Extension>
      <TypeInfo>      </TypeInfo>
    </Extension>
  </ws_object>
</data>

```

JSON Request

```

{
  "data": {
    "ws_object": {
      "Extension": {
        "@GUID": "<guidOfextension>",
        "Extension": " ",
        "TypeInfo": " "
      }
    }
  }
}

```



```

    }
  }
}

```

NOTE: <ws_object> block can be repeated to update multiple users in a single request.

XSDs: DATA, WS_OBJECT, EXTENSION

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.

Sample code:

Refer section [Sample code for PUT](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/extensions>

Output Parameters:

An XML document indicating the success or failure of the update request.

XSDs: RESPONSE, DATA, WS_OBJECT, EXTENSION, ERROR

Example 1: Update an extension in the system – failure – No GUID

```

PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/extensions HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=9413A0345BE1E59DC0EA0DB80467FAE9
X-User-Agent: Avaya-SDKUser
Content-Length: 127
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

```

XML Request

```

<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <Extension>
      <TypeInfo>1</TypeInfo>
    </Extension>
  </ws_object>
</data>

```

JSON Request

```

{
  "data": {
    "ws_object": {
      "Extension": { "TypeInfo": "1" }
    }
  }
}

```

```
}
}
}
```

Result: Indicates an error with the update request. The error indicates that no GUID was supplied to identify the user.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Fri, 27 Nov 2015 14:42:07 GMT
Content-Type: application/xml
Content-Length: 434
Date: Fri, 27 Nov 2015 14:42:05 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <EXTENSION>
        <error>
          <error_code>15</error_code>
          <error_desc>"GUID missing in body"</error_desc>
          <cause>0</cause>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </EXTENSION>
    </ws_object>
    <ws_object error_node="true">
      <REBOOT>
        <error>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </REBOOT>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": [
        {
          "@error_node": "true",
          "EXTENSION": {
            "error": {
              "error_code": "15",
              "error_desc": "\"GUID missing in body\"",
              "cause": "0",
              "errorLocation": "10.240.0.5"
            }
          }
        },
        {
          "@error_node": "true",
          "REBOOT": {
            "error": { "errorLocation": "10.240.0.5" }
          }
        }
      ]
    }
  }
}
```

```
}
}
}
}
```

Example 2: Update existing extension type information - success

```
PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/extensions HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Cookie: JSESSIONID=9413A0345BE1E59DC0EA0DB80467FAE9
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Content-Length: 167
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

XML Request

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <Extension GUID="BE6DAD809AFB11DEB6F242010AF00005">
      <TypeInfo>1</TypeInfo>
    </Extension>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "Extension": {
        "@GUID": "BE6DAD809AFB11DEB6F242010AF00005",
        "TypeInfo": "1"
      }
    }
  }
}
```

Result: Indicates a successful update of the extension.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Fri, 27 Nov 2015 14:33:16 GMT
Content-Type: application/xml
Content-Length: 263
Date: Fri, 27 Nov 2015 14:33:14 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <Extension GUID="BE6DAD809AFB11DEB6F242010AF00005"/>
    </ws_object>
  </data>
</response>
```

```
<ModifiedEntities/>
</ws_object>
<ws_object>
  <reboot type="normal"/>
</ws_object>
</data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
          "Extension": { "@GUID": "BE6DAD809AFB11DEB6F242010AF00005" }
        },
        {
          "reboot": { "@type": "normal" }
        }
      ]
    }
  }
}
```

2.8.5.4 Extensions (DELETE)

Description: REST service which deletes an existing extension on a system.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/extensions

Method: DELETE

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.
- Guid (required) – GUID of the extension to delete from the system.

Sample code:

Refer section [Sample code for DELETE](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/extensions>

Output Parameters:

An XML document indicating the success or failure of the delete request.

XSDs: RESPONSE, DATA, WS_OBJECT, EXTENSION, ERROR

Example 1: Delete an extension in the system – failure – No GUID

```
DELETE https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/extensions HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=9413A0345BE1E59DC0EA0DB80467FAE9
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result: Indicates an error with the delete request. The reason was no GUID was supplied.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Fri, 27 Nov 2015 14:47:30 GMT
Content-Type: application/xml
Content-Length: 307
Date: Fri, 27 Nov 2015 14:47:28 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>400:BadRequest</error_code>
          <error_desc>10.240.0.5</error_desc>
          <cause/>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
```

```
DELETE https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/extensions?guid=1234
HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=9413A0345BE1E59DC0EA0DB80467FAE9
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result: Indicates an error with the delete request. The reason was no GUID was supplied.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Fri, 27 Nov 2015 14:50:32 GMT
Content-Type: application/xml
Content-Length: 305
Date: Fri, 27 Nov 2015 14:50:30 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>404:NotFound</error_code>
          <error_desc>10.240.0.5</error_desc>
          <cause/>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
            "error_code": "404:NotFound",
            "error_desc": "10.240.0.5",

```

```

        "errorLocation": "10.240.0.5"
      }
    }
  }
}

```

Example 3: Delete an existing extension - success

```

DELETE
https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/extensions?guid=BE6DAD809AFB11D
EB6F242010AF00005 HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=9413A0345BE1E59DC0EA0DB80467FAE9
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

```

Result: Indicates a successful deletion of the extension.

```

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Fri, 27 Nov 2015 14:53:58 GMT
Content-Type: application/xml
Content-Length: 188
Date: Fri, 27 Nov 2015 14:53:56 GMT

```

XML Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <ModifiedEntities/>
    </ws_object>
    <ws_object>
      <reboot type="normal"/>
    </ws_object>
  </data>
</response>

```

JSON Response

```

{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
        },
        {
          "reboot": { "@type": "normal" }
        }
      ]
    }
  }
}

```

2.8.6 HuntGroups

2.8.6.1 HuntGroups (GET)

Description: REST service which queries/lists huntgroups on a solution or node in a solution.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/huntgroups>

Method: GET

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- **ipaddress** (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/huntgroups>

Output Parameters:

An XML document containing information on each extension defined in the solution or solution node including (but not limited to):

- **Response Status** – A value of 1 indicates Success, 0 is a failure
- **Total Number of Extension Records**
- **GUID** – Global Unique Identifier which uniquely identifies the huntgroup in the system.
- **Name** – The name of the huntgroup
- **Extension** – extension number of the huntgroup
- **MembersList** – A '|' separated list of extensions in the huntgroup.
- **GroupRingMode** – The type of the huntgroup(Sequential, Cyclical, ...):
 - Collective = 0
 - Sequential = 1
 - Rotary = 2

- Longest Waiting = 3
- Collective Call Waiting = 4

XSDs: RESPONSE, DATA, WS_OBJECT, HUNTGROU, ERROR

Example 1: Get all huntgroups defined in the solution – Error Session Timeout

```
GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/huntgroups HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=6FBDC5BED65E340D50DFE09A703F8973
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

```
HTTP/1.1 461
Server: Apache-Coyote/1.1
Content-Length: 304
Date: Fri, 27 Nov 2015 14:57:33 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>461:SessionTimeout</error_code>
          <error_desc>130.211.136.50</error_desc>
          <errorLocation>130.211.136.50</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
            "error_code": "461:SessionTimeout",
            "error_desc": "130.211.136.50",
            "errorLocation": "130.211.136.50"
          }
        }
      }
    }
  }
}
```

Example 2: Get all huntgroups defined in the solution – Success

GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/huntgroups HTTP/1.1
 Accept-Encoding: gzip,deflate
 Cookie: JSESSIONID=208E77B11EAE3CA8ECCF41DB73FB9827
 X-User-Client: Avaya-WebAdmin
 X-User-Agent: Avaya-SDKUser
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Fri, 27 Nov 2015 15:00:32 GMT
 Content-Type: application/xml
 Content-Length: 441
 Date: Fri, 27 Nov 2015 15:00:30 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data set="75" totalRecords="1">
    <ws_object>
      <HuntGroup GUID="6E7450809AC711DEAE4242010AF00005">
        <Name>SalesForce</Name>
        <Extension>202</Extension>
        <GroupRingMode>1</GroupRingMode>
        <MembersList>205|305|208|321|</MembersList>
        <ServiceMode>0</ServiceMode>
        <VoicemailOn>true</VoicemailOn>
        <IsGroupDistributed>false</IsGroupDistributed>
      </HuntGroup>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "@set": "75",
      "@totalRecords": "1",
      "ws_object": {
        "HuntGroup": {
          "@GUID": "6E7450809AC711DEAE4242010AF00005",
          "Name": "SalesForce",
          "Extension": "202",
          "GroupRingMode": "1",
          "MembersList": "205|305|208|321|",
          "ServiceMode": "0",
          "VoicemailOn": "true",
          "IsGroupDistributed": "false"
        }
      }
    }
  }
}
```

2.8.6.2 HuntGroups (POST)

Description: REST service which adds a huntgroup to a solution node.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/huntgroups

Method: POST

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Content-Length: <length in bytes of the xml body>

Input Body:

Content body of POST request must be of the following format:

XML Request

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <HuntGroup>
      <Name>          </Name>
      <Extension>     </Extension>
      <MembersList>   </MembersList>
      <VoicemailOn>   </VoicemailOn>
      <ServiceMode>   </ServiceMode>
      <GroupRingMode> </GroupRingMode>
    </HuntGroup>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "HuntGroup": {
        "Name": " ",
        "Extension": " ",
        "MembersList": " ",
        "VoicemailOn": " ",
        "ServiceMode": " ",
        "GroupRingMode": " "
      }
    }
  }
}
```

NOTE: <ws_object> block can be repeated to add multiple huntgroups in a single request.

XSDs: DATA, WS_OBJECT, HUNTGROUP

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.

Sample code:

Refer section [Sample code for POST](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/huntgroups>

Output Parameters:

An XML document indicating the success or failure of the Add request.

XSDs: RESPONSE, DATA, WS_OBJECT, HUNTGROUP, ERROR

Example 1: Add 1 new Longest Waiting huntgroup to a system with 4 members - success

```
POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/huntgroups HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=208E77B11EAE3CA8ECCF41DB73FB9827
X-User-Agent: Avaya-SDKUser
Content-Length: 289
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

XML Request

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <HuntGroup>
      <Name>SupportGroup</Name>
      <Extension>407</Extension>
      <MembersList>301|393|302</MembersList>
      <VoicemailOn>true</VoicemailOn>
      <ServiceMode/>
      <GroupRingMode>3</GroupRingMode>
    </HuntGroup>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "HuntGroup": {
        "Name": "SupportGroup",
        "Extension": "407",
        "MembersList": "301|393|302",
        "VoicemailOn": "true",
        "GroupRingMode": "3"
      }
    }
  }
}
```

Result: Indicates success and returns the GUID of the newly created huntgroup.

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Fri, 27 Nov 2015 15:13:25 GMT
 Content-Type: application/xml
 Content-Length: 278
 Date: Fri, 27 Nov 2015 15:13:23 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <HuntGroup GUID="3d7bd1809c6011debd5442010af00005">
        <IsGroupDistributed>false</IsGroupDistributed>
      </HuntGroup>
    </ws_object>
    <ws_object>
      <reboot type="normal"/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
          "HuntGroup": {
            "@GUID": "3d7bd1809c6011debd5442010af00005",
            "IsGroupDistributed": "false"
          }
        },
        {
          "reboot": { "@type": "normal" }
        }
      ]
    }
  }
}
```

Example 2: Add 1 new huntgroup to a system – failure – name already exists

POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/huntgroups HTTP/1.1
 Accept-Encoding: gzip,deflate
 Content-Type: application/xml
 Cookie: JSESSIONID=208E77B11EAE3CA8ECCF41DB73FB9827
 X-User-Client: Avaya-WebAdmin
 X-User-Agent: Avaya-SDKUser
 Content-Length: 289
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <HuntGroup>
      <Name>SupportGroup</Name>
      <Extension>408</Extension>
      <MembersList>301|303|302</MembersList>
      <VoicemailOn>true</VoicemailOn>
      <ServiceMode/>
      <GroupRingMode>2</GroupRingMode>
    </HuntGroup>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "HuntGroup": {
        "Name": "SupportGroup",
        "Extension": "408",
        "MembersList": "301|303|302",
        "VoicemailOn": "true",
        "GroupRingMode": "2"
      }
    }
  }
}
```

Result: Indicates an error because the name already exists.

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Fri, 27 Nov 2015 15:23:12 GMT
 Content-Type: application/xml
 Content-Length: 465
 Date: Fri, 27 Nov 2015 15:23:10 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <HUNTGROUP>
        <error>
          <error_code>9022</error_code>
          <error_desc>"The name already exists."</error_desc>
          <cause>0</cause>
          <cause_data>0</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </HUNTGROUP>
    </ws_object>
    <ws_object error_node="true">
      <REBOOT>
        <error>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </REBOOT>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": [
        {
          "@error_node": "true",
          "HUNTGROUP": {
            "error": {
              "error_code": "9022",
              "error_desc": "\"The name already exist.\"\"",
              "cause": "0",
              "cause_data": "0",
              "errorLocation": "10.240.0.5"
            }
          }
        },
        {
          "@error_node": "true",
          "REBOOT": {
            "error": { "errorLocation": "10.240.0.5" }
          }
        }
      ]
    }
  }
}
```

Example 3: Add 1 new huntgroup to a system – failure – extension already used.

```
POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/huntgroups HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=208E77B11EAE3CA8ECCF41DB73FB9827
X-User-Agent: Avaya-SDKUser
Content-Length: 290
Host: 130.211.136.50:7070
Connection: Keep-Alive
```

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <HuntGroup>
      <Name>SupportGroup2</Name>
      <Extension>301</Extension>
      <MembersList>304|303|302</MembersList>
      <VoicemailOn>true</VoicemailOn>
      <ServiceMode/>
      <GroupRingMode>2</GroupRingMode>
    </HuntGroup>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "HuntGroup": {
        "Name": "SupportGroup2",
        "Extension": "301",
        "MembersList": "304|303|302",
        "VoicemailOn": "true",
        "GroupRingMode": "2"
      }
    }
  }
}
```

Result: Indicates an error because the extension is already used.

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Fri, 27 Nov 2015 15:26:04 GMT
 Content-Type: application/xml
 Content-Length: 509
 Date: Fri, 27 Nov 2015 15:26:02 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <HUNTGROUP>
        <error>
          <error_code>10003</error_code>
          <error_desc>"Huntgroup extension number "301" is used by another
user"</error_desc>
          <cause>0</cause>
          <cause_data>0</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </HUNTGROUP>
    </ws_object>
    <ws_object error_node="true">
      <REBOOT>
        <error>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </REBOOT>
    </ws_object>
  </data>
</response>
```



```
</REBOOT>
</ws_object>
</data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": [
        {
          "@error_node": "true",
          "HUNTGROU": {
            "error": {
              "error_code": "10003",
              "error_desc": "\"Huntgroup extension number \"301\" is used by another user\"",
              "cause": "0",
              "cause_data": "0",
              "errorLocation": "10.240.0.5"
            }
          }
        },
        {
          "@error_node": "true",
          "REBOOT": {
            "error": { "errorLocation": "10.240.0.5" }
          }
        }
      ]
    }
  }
}
```

2.8.6.3 HuntGroups (PUT)

Description: REST service which updates an existing huntgroup on a system.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/huntgroups>

Method: PUT

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Content-Length: <length in bytes of the xml body>

Input Body:

Content body of PUT request must be of the following format (note that the GUID of the huntgroup to update must be included as an attribute of the HuntGroup):

XML Request Template

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <HuntGroup GUID="<guid of huntgroup>">
      <Name>      </Name>
      <Extension> </Extension>
      <MembersList> </MembersList>
      <VoicemailOn> </VoicemailOn>
      <ServiceMode> </ServiceMode>
      <GroupRingMode> </GroupRingMode>
    </HuntGroup>
  </ws_object>
</data>
```

JSON Request Template

```
{
  "data": {
    "ws_object": {
      "HuntGroup": {
        "@GUID": "<guid of huntgroup>",
        "Name": " ",
        "Extension": " ",
        "MembersList": " ",
        "VoicemailOn": " ",
        "ServiceMode": " ",
        "GroupRingMode": " "
      }
    }
  }
}
```

NOTE: <ws_object> block can be repeated to update multiple huntgroups in a single request.

XSDs: DATA, WS_OBJECT, HUNTGROU

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.

Sample code:

Refer section [Sample code for PUT](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/huntgroups>

Output Parameters:

An XML document indicating the success or failure of the update request.

XSDs: RESPONSE, DATA, WS_OBJECT, HUNTGROU, ERROR

Example 1: Update a huntgroup in the system – failure – Invalid GUID

```
PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/huntgroups HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=208E77B11EAE3CA8ECCF41DB73FB9827
X-User-Agent: Avaya-SDKUser
Content-Length: 303
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

XML Request

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <HuntGroup GUID="12345">
      <Name>SupportGroup2</Name>
      <Extension>410</Extension>
      <MembersList>307|303|302</MembersList>
      <VoicemailOn>true</VoicemailOn>
      <ServiceMode/>
      <GroupRingMode>2</GroupRingMode>
    </HuntGroup>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "HuntGroup": {
        "@GUID": "12345",
        "Name": "SupportGroup2",
        "Extension": "410",
        "MembersList": "307|303|302",
        "VoicemailOn": "true",
        "GroupRingMode": "2"
      }
    }
  }
}
```

Result: Indicates an error with the update request. The error indicates that an invalid GUID was supplied to identify the huntgroup.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Fri, 27 Nov 2015 15:55:28 GMT
Content-Type: application/xml
Content-Length: 465
Date: Fri, 27 Nov 2015 15:55:26 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <HUNTGROU>
        <error>
          <error_code>10003</error_code>
          <error_desc>"Object does not exists"</error_desc>
          <cause>0</cause>
          <cause_data>12345</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </HUNTGROU>
    </ws_object>
    <ws_object error_node="true">
      <REBOOT>
        <error>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </REBOOT>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": [
        {
          "@error_node": "true",
          "HUNTGROU": {
            "error": {
              "error_code": "10003",
              "error_desc": "\"Object does not exists\"",
              "cause": "0",
              "cause_data": "12345",
              "errorLocation": "10.240.0.5"
            }
          }
        },
        {
          "@error_node": "true",
          "REBOOT": {
            "error": { "errorLocation": "10.240.0.5" }
          }
        }
      ]
    }
  }
}
```

Example 2: Update existing huntgroup information – success

Almost all data on the huntgroup is changed/updated including the name, extension, members list.... The original data for this huntgroup was:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <HuntGroup GUID="6E7450809AC711DEAE4242010AF00005">
        <Name>SalesForce</Name>
        <Extension>202</Extension>
        <GroupRingMode>1</GroupRingMode>
        <MembersList>205|305|208|321|</MembersList>
        <ServiceMode>0</ServiceMode>
        <VoicemailOn>true</VoicemailOn>
        <IsGroupDistributed>false</IsGroupDistributed>
      </HuntGroup>
    </ws_object>
  </data>
</response>
```

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": {
        "HuntGroup": {
          "@GUID": "6E7450809AC711DEAE4242010AF00005",
          "Name": "SalesForce",
          "Extension": "202",
          "GroupRingMode": "1",
          "MembersList": "205|305|208|321|",
          "ServiceMode": "0",
          "VoicemailOn": "true",
          "IsGroupDistributed": "false"
        }
      }
    }
  }
}
```

PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/huntgroups HTTP/1.1
 Accept-Encoding: gzip,deflate
 Content-Type: application/xml
 Cookie: JSESSIONID=208E77B11EAE3CA8ECCF41DB73FB9827
 X-User-Client: Avaya-WebAdmin
 X-User-Agent: Avaya-SDKUser
 Content-Length: 330
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <HuntGroup GUID="6E7450809AC711DEAE4242010AF00005">
      <Name>SupportGroup2</Name>
      <Extension>410</Extension>
      <MembersList>307|303|302</MembersList>
      <VoicemailOn>true</VoicemailOn>
      <ServiceMode/>
      <GroupRingMode>2</GroupRingMode>
    </HuntGroup>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "HuntGroup": {
        "@GUID": "6E7450809AC711DEAE4242010AF00005",
        "Name": "SupportGroup2",
        "Extension": "410",
        "MembersList": "307|303|302",
        "VoicemailOn": "true",
        "GroupRingMode": "2"
      }
    }
  }
}
```

Result: Indicates a successful update of the huntgroup.

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Fri, 27 Nov 2015 15:45:23 GMT
 Content-Type: application/xml
 Content-Length: 320
 Date: Fri, 27 Nov 2015 15:45:21 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <HuntGroup GUID="6E7450809AC711DEAE4242010AF00005">
        <IsGroupDistributed>false</IsGroupDistributed>
      </HuntGroup>
    </ws_object>
    <ws_object>
      <ModifiedEntities/>
    </ws_object>
    <ws_object>
      <reboot type="normal"/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
          "HuntGroup": {
            "@GUID": "6E7450809AC711DEAE4242010AF00005",
            "IsGroupDistributed": "false"
          }
        },
        {
          "reboot": { "@type": "normal" }
        }
      ]
    }
  }
}
```

```
}
}
}
```

2.8.6.4 HuntGroups (DELETE)

Description: REST service which deletes an existing huntgroup on a system.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/huntgroups>

Method: DELETE

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.
- Guid (required) – GUID of the huntgroup to delete from the system.

Sample code:

Refer section [Sample code for DELETE](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/huntgroups>

Output Parameters:

An XML document indicating the success or failure of the delete request.

XSDs: RESPONSE, DATA, WS_OBJECT, HUNTGROUP, ERROR

Example 1: Delete a huntgroup in the system – failure – Not a Huntgroup GUID

```
DELETE
https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/huntgroups?guid=BE6DAD809AFB11
DEB6F242010AF00005 HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=A94CB476BF7B8A1E1821AB68BEEF582C
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
```

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Result: Indicates an error with the delete request. The reason was that it was not a huntgroup GUID that was supplied.

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Fri, 27 Nov 2015 16:03:54 GMT
 Content-Type: application/xml
 Content-Length: 305
 Date: Fri, 27 Nov 2015 16:03:52 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>404:NotFound</error_code>
          <error_desc>10.240.0.5</error_desc>
          <cause/>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
            "error_code": "404:NotFound",
            "error_desc": "10.240.0.5",
            "errorLocation": "10.240.0.5"
          }
        }
      }
    }
  }
}
```

Example 2: Delete an existing huntgroup - success

DELETE
 https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/huntgroups?guid=6E7450809AC711D
 EAE4242010AF00005 HTTP/1.1
 Accept-Encoding: gzip,deflate
 X-User-Client: Avaya-WebAdmin
 Cookie: JSESSIONID=A94CB476BF7B8A1E1821AB68BEEF582C
 X-User-Agent: Avaya-SDKUser
 Host: 130.211.136.50:7070
 Connection: Keep-Alive

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Result: Indicates a successful deletion of the huntgroup.

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Fri, 27 Nov 2015 16:07:27 GMT
 Content-Type: application/xml
 Content-Length: 188
 Date: Fri, 27 Nov 2015 16:07:25 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <ModifiedEntities/>
    </ws_object>
    <ws_object>
      <reboot type="normal"/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
        },
        {
          "reboot": { "@type": "normal" }
        }
      ]
    }
  }
}
```

2.8.7 ShortCodes

2.8.7.1 ShortCodes (GET)

Description: REST service which queries/lists all shortcodes on a solution or node in a solution.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/shortcodes

Method: GET

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/shortCodes>

Output Parameters:

An XML document containing information on each shortcode defined in the solution or solution node including (but not limited to):

- Response Status – A value of 1 indicates Success, 0 is a failure
- Total Number of ShortCode Records
- *GUID* – Global Unique Identifier which uniquely identifies the shortcode in the system.
- Code
- TelephoneNumber
- LineGroupId to use
- Feature Number
- ForceAccountCode boolean indicating whether or not an account code is required.

XSDs: RESPONSE, DATA, WS_OBJECT, SHORTCODE, ERROR

Example 1: Get all shortcodes defined in the solution.

```
GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/shortcodes HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Tue, 08 Dec 2015 13:39:29 GMT
 Content-Type: application/xml
 Content-Length: 362
 Date: Tue, 08 Dec 2015 13:39:27 GMT

XML Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data set="75" totalRecords="1">
    <ws_object>
      <ShortCode GUID="1F1DB000A43311DE884842010AF00005">
        <Code>XXXXXXXXXX</Code>
        <TelephoneNumber>N</TelephoneNumber>
        <LineGroupId>1</LineGroupId>
        <Feature>31</Feature>
        <ForceAccountCode>false</ForceAccountCode>
      </ShortCode>
    </ws_object>
  </data>
</response>
  
```

JSON Response

```

{
  "response": {
    "@status": "1",
    "data": {
      "@set": "75",
      "@totalRecords": "1",
      "ws_object": {
        "ShortCode": {
          "@GUID": "1F1DB000A43311DE884842010AF00005",
          "Code": "XXXXXXXXXX",
          "TelephoneNumber": "N",
          "LineGroupId": "1",
          "Feature": "31",
          "ForceAccountCode": "false"
        }
      }
    }
  }
}
  
```

2.8.7.2 ShortCodes (PUT)

Description: REST service which updates an existing shortcode on a solution or node in a solution.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/shortcodes>

Method: PUT

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.
- Content body of PUT request must be of the following format (note that the GUID of the shortcode to update must be included as an attribute of the ShortCode):

XML Request Template

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <ShortCode GUID="<GUID of ShortCode to update>">
      <Code/>
      <TelephoneNumber>      </TelephoneNumber>
      <LineGroupId>          </LineGroupId>
      <Feature>              </Feature>
      <ForceAccountCode>     </ForceAccountCode>
    </ShortCode>
  </ws_object>
</data>
```

JSON Request Template

```
{
  "data": {
    "ws_object": {
      "ShortCode": {
        "-GUID": "<GUID of ShortCode to update>",
        "TelephoneNumber": "      ",
        "LineGroupId": "          ",
        "Feature": "              ",
        "ForceAccountCode": "     "
      }
    }
  }
}
```

NOTE: <ws_object> block can be repeated to update multiple shortcodes in a single request.

XSDs: DATA, WS_OBJECT, SHORTCODE

Sample code:

Refer section [Sample code for PUT](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/shortcodes>

Output Parameters:

An XML document containing information on the success or failure of the shortcode update.

- Response Status – A value of 1 indicates Success, 0 is a failure

XSDs: RESPONSE, DATA, ERROR

Example 1: Update an existing shortcode defined in the solution (success).
Actual Code changed from XXXXXXXXXXXX to XXXXXXXXXXXX1

```
PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/shortcodes HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Content-Length: 392
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <ShortCode GUID="1F1DB000A43311DE884842010AF00005">
      <Code>XXXXXXXXXX1</Code>
      <TelephoneNumber>N</TelephoneNumber>
      <LineGroupId>1</LineGroupId>
      <Feature>31</Feature>
      <ForceAccountCode>>false</ForceAccountCode>
    </ShortCode>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "ShortCode": {
        "@GUID": "1F1DB000A43311DE884842010AF00005",
        "Code": "XXXXXXXXXX1",
        "TelephoneNumber": "N",
        "LineGroupId": "1",
        "Feature": "31",
        "ForceAccountCode": "false"
      }
    }
  }
}
```

Result:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
```

Expires: Tue, 08 Dec 2015 14:06:55 GMT
 Content-Type: application/xml
 Content-Length: 94
 Date: Tue, 08 Dec 2015 14:06:53 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data/>
</response>
```

JSON Response

```
{
  "response": { "@status": "1" }
}
```

Example 2: Update an existing shortcode defined in the solution (failure).
 ShortCode GUID is not found.

PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/shortcodes HTTP/1.1
 Accept-Encoding: gzip,deflate
 Content-Type: application/xml
 X-User-Client: Avaya-WebAdmin
 Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
 X-User-Agent: Avaya-SDKUser
 Content-Length: 391
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <ShortCode GUID="1F1DB000A83311DE884842010AF00005">
      <Code>XXXXXXXXXX</Code>
      <TelephoneNumber>N</TelephoneNumber>
      <LineGroupId>1</LineGroupId>
      <Feature>31</Feature>
      <ForceAccountCode>>false</ForceAccountCode>
    </ShortCode>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "ShortCode": {
        "@GUID": "1F1DB000A83311DE884842010AF00005",
        "Code": "XXXXXXXXXX",
        "TelephoneNumber": "N",
        "LineGroupId": "1",
        "Feature": "31",
        "ForceAccountCode": "false"
      }
    }
  }
}
```

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Tue, 08 Dec 2015 14:42:25 GMT
 Content-Type: application/xml
 Content-Length: 341
 Date: Tue, 08 Dec 2015 14:42:23 GMT

XML Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <SHORTCODE>
        <error>
          <error_code>6</error_code>
          <error_desc>"Object not found"</error_desc>
          <cause>0</cause>
          <cause_data>0</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </SHORTCODE>
    </ws_object>
  </data>
</response>
  
```

JSON Response

```

{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SHORTCODE": {
          "error": {
            "error_code": "6",
            "error_desc": "\"Object not found\"",
            "cause": "0",
            "cause_data": "0",
            "errorLocation": "10.240.0.5"
          }
        }
      }
    }
  }
}
  
```

Example 3: Update an existing shortcode defined in the solution (failure).
 Feature invalid.

PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/shortcodes HTTP/1.1
 Accept-Encoding: gzip,deflate
 Content-Type: application/xml
 Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
 X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser
 Content-Length: 393
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <ShortCode GUID="1F1DB000A43311DE884842010AF00005">
      <Code>XXXXXXXXXX</Code>
      <TelephoneNumber>N</TelephoneNumber>
      <LineGroupId>1</LineGroupId>
      <Feature>3132</Feature>
      <ForceAccountCode>>false</ForceAccountCode>
    </ShortCode>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "ShortCode": {
        "@GUID": "1F1DB000A43311DE884842010AF00005",
        "Code": "XXXXXXXXXX",
        "TelephoneNumber": "N",
        "LineGroupId": "1",
        "Feature": "3132",
        "ForceAccountCode": "false"
      }
    }
  }
}
```

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Tue, 08 Dec 2015 14:46:18 GMT
 Content-Type: application/xml
 Content-Length: 332
 Date: Tue, 08 Dec 2015 14:46:16 GMT

XML Response


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <SHORTCODE>
        <error>
          <error_code>2</error_code>
          <error_desc>"Feature"</error_desc>
          <cause>0</cause>
          <cause_data>0</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </SHORTCODE>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SHORTCODE": {
          "error": {
            "error_code": "2",
            "error_desc": "\"Feature\"",
            "cause": "0",
            "cause_data": "0",
            "errorLocation": "10.240.0.5"
          }
        }
      }
    }
  }
}
```

2.8.7.3 ShortCodes (POST)

Description: REST service which adds a shortcode(s) on a solution or node in a solution.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/shortcodes>

Method: POST

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- **ipaddress** (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the

entire solution.

- Content body of POST request must be of the following format :

XML Request Template

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <ShortCode>
      <Code>      </Code>
      <TelephoneNumber>    </TelephoneNumber>
      <LineGroupId>    </LineGroupId>
      <Feature>    </Feature>
      <ForceAccountCode> </ForceAccountCode>
    </ShortCode>
  </ws_object>
</data>
```

JSON Request Template

```
{
  "data": {
    "ws_object": {
      "ShortCode": {
        "Code": " ",
        "TelephoneNumber": " ",
        "LineGroupId": " ",
        "Feature": " ",
        "ForceAccountCode": " "
      }
    }
  }
}
```

NOTE: <ws_object> block can be repeated to add multiple shortcodes in a single request.

XSDs: DATA, WS_OBJECT, SHORTCODE

Sample code:

Refer section [Sample code for POST](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/extensions>

Output Parameters:

XML document containing information on the success or failure of the shortcode adds.

- Response Status – A value of 1 indicates Success, 0 is a failure
- ShortCode GUID(s) on success.

XSDs: RESPONSE, DATA, SHORTCODE, ERROR

Example 1: Add two new shortcodes to a solution (success). One shortcode

has a code of 8XXXXXXXXXX and the other a code of 81XXXXX.

```
POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/shortcodes HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Content-Length: 641
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <ShortCode>
      <Code>8XXXXXXXXXX</Code>
      <TelephoneNumber>N</TelephoneNumber>
      <LineGroupId>2</LineGroupId>
      <Feature>31</Feature>
      <ForceAccountCode>false</ForceAccountCode>
    </ShortCode>
  </ws_object>
  <ws_object>
    <ShortCode>
      <Code>81XXXXX</Code>
      <TelephoneNumber>N</TelephoneNumber>
      <LineGroupId>1</LineGroupId>
      <Feature>31</Feature>
      <ForceAccountCode>true</ForceAccountCode>
    </ShortCode>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": [
      {
        "ShortCode": {
          "Code": "8XXXXXXXXX",
          "TelephoneNumber": "N",
          "LineGroupId": "2",
          "Feature": "31",
          "ForceAccountCode": "false"
        }
      },
      {
        "ShortCode": {
          "Code": "81XXXXX",
          "TelephoneNumber": "N",
          "LineGroupId": "1",
          "Feature": "31",
          "ForceAccountCode": "true"
        }
      }
    ]
  }
}
```

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Tue, 08 Dec 2015 14:34:38 GMT
 Content-Type: application/xml
 Content-Length: 250
 Date: Tue, 08 Dec 2015 14:34:36 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <ShortCode GUID="A46E2380A4FF11DEB4A542010AF00005"/>
    </ws_object>
    <ws_object>
      <ShortCode GUID="A5066F00A4FF11DEB4A542010AF00005"/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
          "ShortCode": { "@GUID": "A46E2380A4FF11DEB4A542010AF00005" }
        },
        {
          "ShortCode": { "@GUID": "A5066F00A4FF11DEB4A542010AF00005" }
        }
      ]
    }
  }
}
```

2.8.7.4 ShortCodes (DELETE)

Description: REST service which deletes an existing Short Code on a system.

URI: <https://{host}:7070/WebManagement/ws/sdk/admin/v1/shortCodes>

Method: DELETE

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.
- Guid (required) – GUID of the Short Code to delete from the system.

Sample code:

Refer section [Sample code for DELETE](#) method to get the system data. The URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/shortCodes>

Output Parameters:

An XML document indicating the success or failure of the delete request.

XSDs: RESPONSE, DATA, WS_OBJECT, ERROR

Example 1: Delete a Short Code from the system – failure – No GUID

```
DELETE https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/shortCodes HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=A5CF980E5986337F884CAD1221D0FE18
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result: Indicates an error with the delete request. The reason was no GUID was supplied.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 22 Dec 2015 18:16:53 GMT
Content-Type: application/xml
Content-Length: 305
Date: Tue, 22 Dec 2015 18:16:51 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>404:NotFound</error_code>
          <error_desc>10.240.0.5</error_desc>
          <cause/>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
            "error_code": "404:NotFound",
            "error_desc": "10.240.0.5",
            "errorLocation": "10.240.0.5"
          }
        }
      }
    }
  }
}
```

Example 2: Delete an existing Short Codes - success

DELETE

```
https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/shortCodes?guid=8CC65800B01F11D
EADD242010AF00005 HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=A5CF980E5986337F884CAD1221D0FE18
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result: Indicates a successful deletion of the Short Code.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 22 Dec 2015 18:24:35 GMT
Content-Type: application/xml
Content-Length: 142
Date: Tue, 22 Dec 2015 18:24:33 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <ModifiedEntities/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
        }
      ]
    }
  }
}
```

2.8.8 Incoming Call Routes (ICR)

2.8.8.1 Incoming Call Routes (GET)

Description: REST service which queries/lists all incoming call routes on a solution or node in a solution.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes

Method: GET

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The URI for consuming this web service should be:

https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes

Output Parameters:

An XML document containing information on each incoming call route defined in the solution or solution node including (but not limited to):

- Response Status – A value of 1 indicates Success, 0 is a failure
- Total Number of Incoming Call Route Records
- *GUID* – Global Unique Identifier which uniquely identifies the incoming call route in the system.
- LineGroupId
- Destination of where to route the call.
- IncomingNumber

XSDs: RESPONSE, DATA, WS_OBJECT, ICR, ERROR

Example 1: Get all incoming call routes defined in the solution (success).

```
GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes
HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Tue, 08 Dec 2015 14:57:05 GMT
 Content-Type: application/xml
 Content-Length: 995
 Date: Tue, 08 Dec 2015 14:57:03 GMT

XML Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data set="75" totalRecords="4">
    <ws_object>
      <IncomingCallRoute GUID="1DC836009AF611DEB51042010AF00005">
        <LineGroupId>50</LineGroupId>
        <CLI/>
        <Destination>3230935</Destination>
        <IncomingNumber>226</IncomingNumber>
      </IncomingCallRoute>
    </ws_object>
    <ws_object>
      <IncomingCallRoute GUID="0111A0809AC011DE9A4242010AF00005">
        <LineGroupId>0</LineGroupId>
        <CLI/>
        <Destination>DialIn</Destination>
        <IncomingNumber/>
      </IncomingCallRoute>
    </ws_object>
    <ws_object>
      <IncomingCallRoute GUID="1DC836009AF611DEB50F42010AF00005">
        <LineGroupId>1</LineGroupId>
        <CLI/>
        <Destination>3230935</Destination>
        <IncomingNumber>226</IncomingNumber>
      </IncomingCallRoute>
    </ws_object>
    <ws_object>
      <IncomingCallRoute GUID="0AE9B9809AC511DEAE0642010AF00005">
        <LineGroupId>2</LineGroupId>
        <CLI/>
        <Destination>VM:Auto Attendant - External</Destination>
        <IncomingNumber>6134545961</IncomingNumber>
      </IncomingCallRoute>
    </ws_object>
  </data>
</response>
  
```

JSON Response

```

{
  "response": {
    "@status": "1",
    "data": {
      "@set": "75",
      "@totalRecords": "4",
      "ws_object": [
        {
          "IncomingCallRoute": {
            "@GUID": "1DC836009AF611DEB51042010AF00005",
            "LineGroupId": "50",
            "Destination": "3230935",
            "IncomingNumber": "226"
          }
        }
      ]
    }
  },
}
  
```

```
{
  "IncomingCallRoute": {
    "@GUID": "0111A0809AC011DE9A4242010AF00005",
    "LineGroupId": "0",
    "Destination": "DialIn"
  }
},
{
  "IncomingCallRoute": {
    "@GUID": "1DC836009AF611DEB50F42010AF00005",
    "LineGroupId": "1",
    "Destination": "3230935",
    "IncomingNumber": "226"
  }
},
{
  "IncomingCallRoute": {
    "@GUID": "0AE9B9809AC511DEAE0642010AF00005",
    "LineGroupId": "2",
    "Destination": "VM:Auto Attendant - External",
    "IncomingNumber": "6134545961"
  }
}
]
}
```

2.8.8.2 Incoming Call Routes (PUT)

Description: REST service which updates an existing incoming call route on a solution or node in a solution.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes

Method: PUT

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.
- Content body of PUT request must be of the following format (note that the GUID of the incoming call route to update must be included as an attribute of the IncomingCallRoute):

XML Request Template

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <IncomingCallRoute GUID="GUID of ICR">
      <LineGroupId>      </LineGroupId>
      <CLI>              </CLI>
      <Destination>      </Destination>
      <IncomingNumber>    </IncomingNumber>
    </IncomingCallRoute>
  </ws_object>
</data>
```

JSON Request Template

```
{
  "data": {
    "ws_object": {
      "IncomingCallRoute": {
        "@GUID": "GUID of ICR",
        "LineGroupId": " ",
        "CLI": " ",
        "Destination": " ",
        "IncomingNumber": " "
      }
    }
  }
}
```

NOTE: <ws_object> block can be repeated to update multiple incoming call routes in a single request.

XSDs: DATA, WS_OBJECT, ICR

Sample code:

Refer section [Sample code for PUT](#) method to get the system data. The URI for consuming this web service should be:

https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes

Output Parameters:

An XML document containing information on the success or failure of the ICR update.

- Response Status – A value of 1 indicates Success, 0 is a failure

XSDs: RESPONSE, DATA, ERROR

Example 1: Update two existing incoming call routes defined in the solution (success).

```
PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes
HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
```

Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
 X-User-Client: Avaya-WebAdmin
 X-User-Agent: Avaya-SDKUser
 Content-Length: 675
 Host: 130.211.136.50:7070
 Connection: Keep-Alive
 User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

XML Request

```

<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <IncomingCallRoute GUID="0AE9B9809AC511DEAE0642010AF00005">
      <LineGroupId>2</LineGroupId>
      <CLI/>
      <Destination>VM:Auto Attendant - External</Destination>
      <IncomingNumber>6134545981</IncomingNumber>
    </IncomingCallRoute>
  </ws_object>
  <ws_object>
    <IncomingCallRoute GUID="1DC836009AF611DEB50F42010AF00005">
      <LineGroupId>1</LineGroupId>
      <CLI/>
      <Destination>3230932</Destination>
      <IncomingNumber>229</IncomingNumber>
    </IncomingCallRoute>
  </ws_object>
</data>
  
```

JSON Request

```

{
  "data": {
    "ws_object": [
      {
        "IncomingCallRoute": {
          "@GUID": "0AE9B9809AC511DEAE0642010AF00005",
          "LineGroupId": "2",
          "Destination": "VM:Auto Attendant - External",
          "IncomingNumber": "6134545981"
        }
      },
      {
        "IncomingCallRoute": {
          "@GUID": "1DC836009AF611DEB50F42010AF00005",
          "LineGroupId": "1",
          "Destination": "3230932",
          "IncomingNumber": "229"
        }
      }
    ]
  }
}
  
```

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Tue, 08 Dec 2015 15:16:01 GMT
 Content-Type: application/xml
 Content-Length: 94

Date: Tue, 08 Dec 2015 15:15:59 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data/>
</response>
```

JSON Response

```
{
  "response": { "-status": "1" }
}
```

Example 2: Update an existing incoming call route defined in the solution (failure). Incoming Call Route GUID is not found.

```
PUT https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes
HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Agent: Avaya-SDKUser
Content-Length: 376
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <IncomingCallRoute GUID="0AD9B9809AC511DEAE0642010AF00005">
      <LineGroupId>2</LineGroupId>
      <CLI/>
      <Destination>VM:Auto Attendant - External</Destination>
      <IncomingNumber>6134545981</IncomingNumber>
    </IncomingCallRoute>
  </ws_object>
</data>
```

JSON Request

```
{
  "data": {
    "ws_object": {
      "IncomingCallRoute": {
        "@GUID": "0AD9B9809AC511DEAE0642010AF00005",
        "LineGroupId": "2",
        "Destination": "VM:Auto Attendant - External",
        "IncomingNumber": "6134545981"
      }
    }
  }
}
```

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Tue, 08 Dec 2015 15:18:38 GMT
 Content-Type: application/xml
 Content-Length: 357
 Date: Tue, 08 Dec 2015 15:18:36 GMT

XML Response

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <INCOMINGCALLROUTE>
        <error>
          <error_code>6</error_code>
          <error_desc>"Object not found"</error_desc>
          <cause>0</cause>
          <cause_data>0</cause_data>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </INCOMINGCALLROUTE>
    </ws_object>
  </data>
</response>
  
```

JSON Response

```

{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "INCOMINGCALLROUTE": {
          "error": {
            "error_code": "6",
            "error_desc": "\"Object not found\"",
            "cause": "0",
            "cause_data": "0",
            "errorLocation": "10.240.0.5"
          }
        }
      }
    }
  }
}
  
```

2.8.8.3 Incoming Call Routes (POST)

Description: REST service which adds a ICR(s) on a solution or node in a solution.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes

Method: POST

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.
- Content body of POST request must be of the following format :

XML Request Template

```
<?xml version='1.0' encoding='UTF-8' ?>
<data>
  <ws_object>
    <IncomingCallRoute>
      <LineGroupId>          </LineGroupId>
      <CLI>                  </CLI>
      <Destination>          </Destination>
      <IncomingNumber>       </IncomingNumber>
    </IncomingCallRoute>
  </ws_object>
</data>
```

JSON Request Template

```
{
  "data": {
    "ws_object": {
      "IncomingCallRoute": {
        "LineGroupId": " ",
        "CLI": " ",
        "Destination": " ",
        "IncomingNumber": " "
      }
    }
  }
}
```

NOTE: <ws_object> block can be repeated to add multiple incoming call routes in a single request.

XSDs: DATA, WS_OBJECT, ICR

Sample code:

Refer section [Sample code for POST](#) method to get the system data. The URI for consuming this web service should be:

https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes

Output Parameters:

An XML document containing information on the success or failure of the

incoming call route which is added.

- Response Status – A value of 1 indicates Success, 0 is a failure
- IncomingCallRoute GUID(s) on success.

XSDs: RESPONSE, DATA, ICR, ERROR

Example 1: Add two incoming call routes to a solution (success).

```
POST https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes
HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: application/xml
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Agent: Avaya-SDKUser
Content-Length: 557
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

XML Request

```
<?xml version='1.0' encoding='UTF-8'?>
<data>
  <ws_object>
    <IncomingCallRoute>
      <LineGroupId>79</LineGroupId>
      <CLI/>
      <Destination>226</Destination>
      <IncomingNumber>123456</IncomingNumber>
    </IncomingCallRoute>
  </ws_object>
  <ws_object>
    <IncomingCallRoute>
      <LineGroupId>2</LineGroupId>
      <CLI/>
      <Destination>227</Destination>
      <IncomingNumber>123457</IncomingNumber>
    </IncomingCallRoute>
  </ws_object>
</data>
```

JSON Request


```
{
  "data": {
    "ws_object": [
      {
        "IncomingCallRoute": {
          "LineGroupId": "79",
          "Destination": "226",
          "IncomingNumber": "123456"
        }
      },
      {
        "IncomingCallRoute": {
          "LineGroupId": "2",
          "Destination": "227",
          "IncomingNumber": "123457"
        }
      }
    ]
  }
}
```

Result:

HTTP/1.1 200 OK
 Server: Apache-Coyote/1.1
 Pragma: no-cache
 Cache-control: max-age=2
 Expires: Tue, 08 Dec 2015 15:27:34 GMT
 Content-Type: application/xml
 Content-Length: 266
 Date: Tue, 08 Dec 2015 15:27:32 GMT

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <IncomingCallRoute GUID="0A120100A50711DEB5E542010AF00005"/>
    </ws_object>
    <ws_object>
      <IncomingCallRoute GUID="0A120100A50711DEB5E642010AF00005"/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
          "IncomingCallRoute": { "@GUID": "0A120100A50711DEB5E542010AF00005" }
        },
        {
          "IncomingCallRoute": { "@GUID": "0A120100A50711DEB5E642010AF00005" }
        }
      ]
    }
  }
}
```

2.8.8.4 Incoming Call Routes (DELETE)

Description: REST service which deletes an existing Incoming Call Route on a system.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes

Method: DELETE

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, add applies to the Primary.
- Guid (required) – GUID of the Short Code to delete from the system.

Sample code:

Refer section [Sample code for DELETE](#) method to get the system data. The URI for consuming this web service should be:

https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes

Output Parameters:

An XML document indicating the success or failure of the delete request.

XSDs: RESPONSE, DATA, WS_OBJECT, ERROR

Example 1: Delete a Incoming Call Route from the system – failure – No GUID

```
DELETE https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes
HTTP/1.1
Accept-Encoding: gzip,deflate
X-User-Client: Avaya-WebAdmin
Cookie: JSESSIONID=A5CF980E5986337F884CAD1221D0FE18
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result: Indicates an error with the delete request. The reason was no

GUID was supplied.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 22 Dec 2015 18:16:53 GMT
Content-Type: application/xml
Content-Length: 305
Date: Tue, 22 Dec 2015 18:16:51 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="0">
  <data>
    <ws_object error_node="true">
      <SMAError>
        <error>
          <error_code>404:NotFound</error_code>
          <error_desc>10.240.0.5</error_desc>
          <cause/>
          <errorLocation>10.240.0.5</errorLocation>
        </error>
      </SMAError>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "0",
    "data": {
      "ws_object": {
        "@error_node": "true",
        "SMAError": {
          "error": {
            "error_code": "404:NotFound",
            "error_desc": "10.240.0.5",
            "errorLocation": "10.240.0.5"
          }
        }
      }
    }
  }
}
```

Example 2: Delete an existing Incoming Call Route - success

```
DELETE
https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/incoming_call_routes?guid=8CC65800
B01F11DEADD242010AF00005 HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=A5CF980E5986337F884CAD1221D0FE18
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result: Indicates a successful deletion of the Incoming Call Route.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 22 Dec 2015 18:24:35 GMT
Content-Type: application/xml
Content-Length: 142
Date: Tue, 22 Dec 2015 18:24:33 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data>
    <ws_object>
      <ModifiedEntities/>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "ws_object": [
        {
        }
      ]
    }
  }
}
```

2.8.9 Incoming Line Group Ids

2.8.9.1 Incoming Line Group Ids (GET)

Description: REST service which queries/lists all incoming line group ids on a solution or node in a solution.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_group_ids

Method: GET

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- ipaddress (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The URI for consuming this web service should be:

https://{host}:7070/WebManagement/ws/sdk/admin/v1/incoming_groups_ids

Output Parameters:

An XML document containing a list of incoming line group ids defined in the solution or solution node including (but not limited to):

- Response Status – A value of 1 indicates Success, 0 is a failure
- Total Number of Incoming Group Ids
- IncomingGroupIdsList – List of incoming line group Ids separated by a '['.
- PBXAddress of system.

XSDs: RESPONSE, DATA, WS_OBJECT, INCOMING_GROUPIDS, ERROR

Example 1: Get a list of incoming line group ids defined in the solution (success).

```
GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/incoming_group_ids
HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 08 Dec 2015 15:41:57 GMT
Content-Type: application/xml
Content-Length: 271
Date: Tue, 08 Dec 2015 15:41:55 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data totalRecords="1">
    <ws_object>
      <IncomingGroupIds>
        <IncomingGroupIdsList>2|8|0|50:Main|</IncomingGroupIdsList>
        <PBXAddress>10.240.0.5</PBXAddress>
      </IncomingGroupIds>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "@totalRecords": "1",
      "ws_object": {
        "IncomingGroupIds": {
          "IncomingGroupIdsList": "2|8|0|50:Main|",
          "PBXAddress": "10.240.0.5"
        }
      }
    }
  }
}
```

2.8.10 Outgoing Line Group Ids

2.8.10.1 Outgoing Line Group Ids (GET)

Description: REST service which queries/lists all outgoing line group ids on a solution or node in a solution.

URI: https://{host}:7070/WebManagement/ws/sdk/admin/v1/outgoing_group_ids

Method: GET

Accept: text/xml

Headers:

X-User-Client: Avaya-WebAdmin

X-User-Agent: Avaya-SDKUser

Content-Type: application/xml OR application/json

Authorization: The username and password are combined into a string with the format "username:password", which is then base64 encoded.

Input Parameters:

- **ipaddress** (optional) – LAN 1 IP Address of the Solution Node on which to apply the command. If not present in the request, query applies to the entire solution.

Sample code:

Refer section [Sample code for GET](#) method to get the system data. The

URI for consuming this web service should be:

<https://{host}:7070/WebManagement/ws/sdk/admin/v1/extensions>

Output Parameters:

An XML document containing a list of outgoing line group ids defined in the solution or solution node including (but not limited to):

- Response Status – A value of 1 indicates Success, 0 is a failure
- Total Number of Outgoing Group Ids records.
- OutgoingGroupIdsList – List of outgoing line group Ids separated by a '|'.
- PBXAddress of system.

XSDs: RESPONSE, DATA, WS_OBJECT, OUTGOING_GROUPIDS, ERROR

Example 1: Get a list of outgoing line group ids defined in the solution (success).

```
GET https://130.211.136.50:7070/WebManagement/ws/sdk/admin/v1/outgoing_group_ids
HTTP/1.1
Accept-Encoding: gzip,deflate
Cookie: JSESSIONID=0A9E6E64F52C74F2445C1E6C8A07FCF6
X-User-Client: Avaya-WebAdmin
X-User-Agent: Avaya-SDKUser
Host: 130.211.136.50:7070
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
```

Result:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Pragma: no-cache
Cache-control: max-age=2
Expires: Tue, 08 Dec 2015 15:43:18 GMT
Content-Type: application/xml
Content-Length: 271
Date: Tue, 08 Dec 2015 15:43:16 GMT
```

XML Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response status="1">
  <data totalRecords="1">
    <ws_object>
      <OutgoingGroupIds>
        <OutgoingGroupIdsList>1|9|0|50:Main|</OutgoingGroupIdsList>
        <PBXAddress>10.240.0.5</PBXAddress>
      </OutgoingGroupIds>
    </ws_object>
  </data>
</response>
```

JSON Response

```
{
  "response": {
    "@status": "1",
    "data": {
      "@totalRecords": "1",
      "ws_object": {
        "OutgoingGroupIds": {
          "OutgoingGroupIdsList": "1|9|0|50:Main|",
          "PBXAddress": "10.240.0.5"
        }
      }
    }
  }
}
```


3 Appendix A – XML Schemas

XML Schema Definitions (i.e. XSD) exists for a number of the XML structures that are used in the REST requests. This section provides information/definition on the XSDs needed in the SDK.

NOTE: Knowledge of XSD language is assumed.

3.1 Common XSDs

Common XSDs are those XML Schema definitions which are common in many requests or responses. Not all of the Common XSDs, however, are included in every request/response XML document.

3.1.1 DATA

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="" xmlns=""
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="ws_object.xsd" />

  <xsd:element name="data">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ws_object" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

3.1.2 WS_OBJECT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace=""
  xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="user.xsd" />
  <xsd:include schemaLocation="reference.xsd" />
  <xsd:include schemaLocation="errorresponse.xsd" />
  <xsd:include schemaLocation="warning.xsd" />

  <xsd:element name="Entity" abstract="true"/>

  <xsd:element name="ws_object">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Entity" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

        <xsd:element ref="reference_list" minOccurs="0" />
        <xsd:element ref="Warning" minOccurs="0" />
    </xsd:sequence>
    <xsd:attribute name="error_node" type="xsd:string" use="optional"/>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

3.1.3 RESPONSE

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="" xmlns=""
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>
  <xsd:include schemaLocation="data.xsd"/>
  <xsd:element name="response">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="data" minOccurs="0" />
      </xsd:sequence>

      <xsd:attribute name="status" type="xsd:string" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="combinedresponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="data" minOccurs="0" maxOccurs="unbounded"/>

      </xsd:sequence>
      <xsd:attribute name="status" type="xsd:string" />
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

3.1.4 REFERENCE

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="" xmlns=""
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="reference_list">
    <xsd:complexType>
      <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ref">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="service" minOccurs="0" maxOccurs="unbounded">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="GUID" type="xsd:string" minOccurs="1"
maxOccurs="1" />
                    <xsd:element name="AccessRights" type="xsd:string" minOccurs="1"
maxOccurs="1" />
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
              <xsd:element name="guidList" minOccurs="0" maxOccurs="unbounded">
                <xsd:complexType>

```

```

        <xsd:sequence>
          <xsd:element name="value" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:simpleContent>
                <xsd:extension base="xsd:string">
                  <xsd:attribute name="name" type="xsd:string" />
                </xsd:extension>
              </xsd:simpleContent>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="name" type="xsd:string" />
</xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

3.1.5 ERROR

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace=""
  xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="error">

    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="error_code" type="xsd:string" />
        <xsd:element name="error_desc" type="xsd:string" />
        <xsd:element name="cause" type="xsd:string" />
        <xsd:element name="cause_data" type="xsd:string" />
        <xsd:element name="errorLocation" type="xsd:string" />
        <xsd:element name="referredMode" type="xsd:string" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

</xsd:schema>

```

3.1.6 WARNING

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="" xmlns=""
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:element name="Warning">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="cause" type="xsd:string" minOccurs="1" maxOccurs="1" />
        <xsd:element name="cause_data" type="xsd:string" minOccurs="1" maxOccurs="1" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

```

```
</xsd:element>
</xsd:schema>
```

3.2 SYSTEM

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="" xmlns=""
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="../common/ws_object.xsd" />
  <xsd:include schemaLocation="../common/error.xsd" />

  <xsd:element name="System" substitutionGroup="Entity">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Name" type="xsd:string" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Password" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="MajorVersion" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="MinorVersion" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="BuildVersion" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="Configuration" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element ref="Telephony" minOccurs="0" maxOccurs="1" />
        <xsd:element ref="Voicemail" minOccurs="0" maxOccurs="1" />
        <xsd:element ref="LANS" minOccurs="0" maxOccurs="1" />
        <xsd:element ref="PBXLicense" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Telephony">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Locale" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="LANS">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="LAN" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="LAN">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="IPAddress" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="IPMask" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
        <xsd:element name="PrimaryTranslationIPAddress" type="xsd:string"
minOccurs="0" maxOccurs="1" />
        <xsd:element name="dhcp_address_number" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="DHCPmode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```

```

        <xsd:element ref="VoipGatekeeper" minOccurs="0" maxOccurs="unbounded" />
        <xsd:element ref="LanDHCPPools" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" />
</xsd:complexType>
</xsd:element>

<xsd:element name="VoipGatekeeper">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="H323GatekeeperEnable" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="H323RemoteExtnEnable" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="H323RemoteExtnCallSignalPort" type="xsd:string"
minOccurs="0" maxOccurs="1" />
            <xsd:element name="H323SignalingOverTLS" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
            <xsd:element name="SipTrunksEnable" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="SipRegistrarEnable" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="SipRemoteExtnEnable" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="DomainName" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="SipUdpEnable" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="UdpPort" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
            <xsd:element name="RemoteUdpPort" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="SipTcpEnable" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="TcpPort" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
            <xsd:element name="RemoteTcpPort" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="SipTlsEnable" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="TlsPort" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
            <xsd:element name="RemoteTlsPort" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="ChallengeExpiryTime" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="RtpPortMin" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="RtpPortMax" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="RtpPortMinNat" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="RtpPortMaxNat" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="RtcpMonitorEnable" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="AutoCreateExtn" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="AutoCreateUser" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="RemoteCallSignallingPort" type="xsd:string"
minOccurs="0" maxOccurs="1" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="Voicemail">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Type" type="xsd:string" minOccurs="0" maxOccurs="1" />

```

```

        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="LanDHCPPools">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="ApplyToAvayaIPPhonesOnly" type="xsd:boolean"
minOccurs="0" maxOccurs="1" />
            <xsd:element ref="DHCPPools" minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="DHCPPools">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="DHCPPool" minOccurs="0" maxOccurs="unbounded" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="DHCPPool">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="DHCPStartAddress" type="xsd:string" />
            <xsd:element name="DHCPSubnetMask" type="xsd:string" />
            <xsd:element name="DHCPDefaultRouter" type="xsd:string" />
            <xsd:element name="DHCPPoolSize" type="xsd:string" />
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" />
    </xsd:complexType>
</xsd:element>

<xsd:element name="PBXLicense">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="EnableRemoteServer" type="xsd:boolean" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="EnableProxyViaPrimaryIPOLine" type="xsd:boolean"
minOccurs="0" maxOccurs="1" />
            <xsd:element name="LicenseServerIPAddress" type="xsd:string"
minOccurs="0" maxOccurs="1" />
            <xsd:element name="LicenseServerURL" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="LicenseServerURLSecondary" type="xsd:string"
minOccurs="0" maxOccurs="1" />
            <xsd:element name="LicenseServerURN" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="LicenseServerPort" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="LicenseServerID" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="LicenseMode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="LicenseGracePeriod" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element ref="ReservedLicenses" minOccurs="0" maxOccurs="1" />
            <xsd:element name="LicensedVersion" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="PLDSHostId" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="PLDSFileStatus" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="SelectLicensingStatus" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="GraceDongleSerialNumber" type="xsd:string"
minOccurs="0" maxOccurs="1" />
            <xsd:element name="DongleSerialGraceDays" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

```

        <xsd:element name="GracePLDSHostId" type="xsd:string" minOccurs="0"
maxOccurs="1" />
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="ReservedLicenses">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="VMProRecordingsAdmins" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="WaveUsers" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="VMProTtsPro" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="VMProTtsWindows" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="VMProTts3rdParty" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="CtiLinkPro" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="EmbeddedVMPorts" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="VMProPorts" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="SipSessions" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="ACCS" type="xsd:string" minOccurs="0" maxOccurs="1" />
            <xsd:element name="BranchSystem" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="SMTrunks" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="UMSWebServices" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="MacSoftphones" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="ThirdPartyRecorder" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="ServerEdition" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="AvayaIPEndpoints" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="ThirdPartyIPEndpoints" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="Receptionist" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="BasicUser" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="OfficeWorker" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="Teleworker" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="MobileUser" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="PowerUser" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="AvayaSoftphone" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="WebCollaboration" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="UniversalPRIAdditionalchannels" type="xsd:string"
minOccurs="0" maxOccurs="1" />
            <xsd:element name="IPSecTunneling" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="CentralizedEndpoints" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="EssentialEdition" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            <xsd:element name="PreferredEdition" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

```

    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

3.3 LICENSE

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace=""
  xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="../../../common/ws_object.xsd" />
  <xsd:include schemaLocation="../../../common/error.xsd" />

  <xsd:element name="Licenses">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="License" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="License" substitutionGroup="Entity">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="LicenseKey" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="Source" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
        <xsd:element name="Type" type="xsd:string" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Status" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
        <xsd:element name="Quantity" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="FreeInstances" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="ExpiryDate" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="Mode" type="xsd:string" minOccurs="0" maxOccurs="1" />
        <xsd:element name="DisplayName" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="AvailableUserPackageLicenseOfType" type="xsd:string"
minOccurs="0" maxOccurs="1" />
        <xsd:element name="IsGrace" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
        <xsd:element name="PBXAddress" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="PowerUserLicensesAvailable" type="xsd:string"
minOccurs="0" maxOccurs="1" />
        <xsd:element ref="error" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name="GUID" type="xsd:string" />
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```


3.4 USER

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="../../common/ws_object.xsd" />
  <xsd:include schemaLocation="../../common/error.xsd" />

  <xsd:element name="User" substitutionGroup="Entity">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Name" type="xsd:string" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Password" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="FullName" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="Extension" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="VoicemailEmail" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="XDirectory" type="xsd:string" default="false"
minOccurs="0" maxOccurs="1" />
        <xsd:element name="SpecificBstType" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="PhoneType" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="VoicemailOn" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="VoicemailCode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="Receptionist" type="xsd:string" default="false"
minOccurs="0" maxOccurs="1" />
        <xsd:element name="AssignedPackage" type="xsd:string" default="1"
minOccurs="0" maxOccurs="1" />
        <xsd:element name="RemoteWorker" type="xsd:string" default="false"
minOccurs="0" maxOccurs="1" />
        <xsd:element name="TwinningType" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="MobilityFeatures" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="UMSWebServices" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="FlareEnabled" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="LoginCode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="SIPName" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
        <xsd:element name="SIPContact" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="WebCollaboration" type="xsd:string" minOccurs="0"
maxOccurs="1"/>
        <xsd:element name="CanIntrude" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="CannotBeIntruded" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="UserRights" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="OutgoingCallBar" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="OutOfHoursUserRights" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="UserRightsTimeProfile" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="Priority" type="xsd:string" minOccurs="0"
maxOccurs="1" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

        </xsd:sequence>
        <xsd:attribute name="GUID" type="xsd:string"/>
    </xsd:complexType>
</xsd:element>
</xsd:schema>

```

3.5 EXTENSION

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="" xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:annotation>
        <xsd:documentation>
        </xsd:documentation>
    </xsd:annotation>

    <xsd:include schemaLocation="../../common/data.xsd" />

    <xsd:element name="Extension" substitutionGroup="Entity">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Id" type="xsd:string" minOccurs="0" maxOccurs="1" />
                <xsd:element name="Extension" type="xsd:string" minOccurs="0"
maxOccurs="1" />
                <xsd:element name="TypeInfo" type="xsd:string" minOccurs="0"
maxOccurs="1" />
                <xsd:element name="Extnclassification" type="xsd:string" minOccurs="0"
maxOccurs="1" />
                <xsd:element name="Port" type="xsd:string" minOccurs="0" maxOccurs="1" />
                <xsd:element name="Module" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
                <xsd:element name="Location" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            </xsd:sequence>
            <xsd:attribute name="GUID" type="xsd:string" />
        </xsd:complexType>
    </xsd:element>
</xsd:schema>

```

3.6 HUNTGROUP

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace=""
    xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <xsd:annotation>
        <xsd:documentation>
        </xsd:documentation>
    </xsd:annotation>

    <xsd:include schemaLocation="../../common/data.xsd" />

    <xsd:element name="HuntGroup" substitutionGroup="Entity">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="Name" type="xsd:string" minOccurs="0" maxOccurs="1" />
                <xsd:element name="Extension" type="xsd:string" minOccurs="0"
maxOccurs="1" />
                <xsd:element name="MembersList" type="xsd:string" minOccurs="0"
maxOccurs="1" />
                <xsd:element name="VoicemailOn" type="xsd:string" minOccurs="0"
maxOccurs="1" />
                <xsd:element name="ServiceMode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
                <xsd:element name="GroupRingMode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>

```

```

        <xsd:element name="NoAnswerTime" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="VoicemailTime" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="VoicemailCode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="GUID" type="xsd:string" />
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

3.7 SHORTCODE

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace=""
  xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>
  <xsd:include schemaLocation="../common/ws_object.xsd" />
  <xsd:include schemaLocation="../common/error.xsd" />
  <xsd:element name="ShortCodes">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ShortCode" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="ShortCode" substitutionGroup="Entity">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Code" type="xsd:string" minOccurs="0" maxOccurs="1" />
        <xsd:element name="TelephoneNumber" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="LineGroupId" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="Feature" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
        <xsd:element name="Locale" type="xsd:string" minOccurs="0" maxOccurs="1"
/>
        <xsd:element name="ForceAccountCode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="ForceAuthCode" type="xsd:string" minOccurs="0"
maxOccurs="1" />
        <xsd:element ref="error" minOccurs="0" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name="GUID" type="xsd:string" />
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

3.8 INCOMINGCALLROUTE (ICR)

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace=""
  xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="../common/ws_object.xsd" />
  <xsd:include schemaLocation="../common/error.xsd" />

```

```

<xsd:element name="IncomingCallRoute" substitutionGroup="Entity">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="LineGroupId" type="xsd:string" minOccurs="0"
maxOccurs="1" />
      <xsd:element name="CLI" type="xsd:string" minOccurs="0" maxOccurs="1" />
      <xsd:element name="Destination" type="xsd:string" minOccurs="0"
maxOccurs="1" />
      <xsd:element name="IncomingNumber" type="xsd:string" minOccurs="0"
maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute name="GUID" type="xsd:string" />
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

3.9 LINE

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace=""
  xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:jaxb="http://java.sun.com/xml/ns/jaxb"
  xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
  jaxb:version="1.0"
>
  <xsd:annotation>
    <xsd:appinfo>
      <jaxb:globalBindings generateIsSetMethod="true">
        <xjc:serializable uid="12343" />
      </jaxb:globalBindings>
    </xsd:appinfo>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="shortcode.xsd" />

  <xsd:element name="Line">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="LineName" type="xsd:string" minOccurs="0" />
        <xsd:element name="LineNumber" type="xsd:string" minOccurs="0" />
        <xsd:element name="TelephoneNumber" type="xsd:string" minOccurs="0" />
        <xsd:element name="IncomingGroupId" type="xsd:string" minOccurs="0" />
        <xsd:element name="LineType" type="xsd:string" minOccurs="0" />
        <xsd:element name="LineSubtype" type="xsd:string" minOccurs="0" />
        <xsd:element name="BChannelSelect" type="xsd:string" minOccurs="0" />
        <xsd:element name="ZeroSuppression" type="xsd:string" minOccurs="0" />
        <xsd:element name="Framing" type="xsd:string" minOccurs="0" />
        <xsd:element name="LineSignalling" type="xsd:string" minOccurs="0" />
        <xsd:element name="ClockSource" type="xsd:string" minOccurs="0" />
        <xsd:element name="IncomingDigits" type="xsd:string" minOccurs="0" />
        <xsd:element name="HaulType" type="xsd:string" minOccurs="0" />
        <xsd:element name="ChannelUnit" type="xsd:string" minOccurs="0" />
        <xsd:element name="CSUOperation" type="xsd:string" minOccurs="0" />
        <xsd:element name="Prefix" type="xsd:string" minOccurs="0" />
        <xsd:element name="NationalPrefix" type="xsd:string" minOccurs="0" />
        <xsd:element name="InternationalPrefix" type="xsd:string" minOccurs="0" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="OutgoingChannels" type="xsd:string" minOccurs="0" />
  <xsd:element name="VoiceChannels" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataChannels" type="xsd:string" minOccurs="0" />
  <xsd:element name="CardOrModule" type="xsd:string" minOccurs="0" />
  <xsd:element name="PortNumber" type="xsd:string" minOccurs="0" />
  <xsd:element name="ChannelsTotal" type="xsd:string" minOccurs="0" />
  <xsd:element name="LineChannels" type="xsd:string" minOccurs="0" />
  <xsd:element name="TEI" type="xsd:string" minOccurs="0" />

```

```

<xsd:element name="CRC" type="xsd:string" minOccurs="0" />
<xsd:element name="LineGroupId" type="xsd:string" minOccurs="0" />
<xsd:element name="LineLength" type="xsd:string" minOccurs="0" />
<xsd:element name="UnitCost" type="xsd:string" minOccurs="0" />
<xsd:element name="CallTracing" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PartialRerouting" type="xsd:boolean" minOccurs="0" />
<xsd:element name="ActiveCCBS" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PassiveCCBS" type="xsd:boolean" minOccurs="0" />
<xsd:element name="USPriTestNumber" type="xsd:string" minOccurs="0" />
<xsd:element name="NetworkTypeIndex" type="xsd:string" minOccurs="0" />
<xsd:element name="NIPIAddition" type="xsd:string" minOccurs="0" />
<xsd:element name="SendRedirectingNumber" type="xsd:boolean"
minOccurs="0" />
<xsd:element name="SendNames" type="xsd:boolean" minOccurs="0" />
<xsd:element name="NamesLength" type="xsd:string" minOccurs="0" />
<xsd:element name="AdminState" type="xsd:string" minOccurs="0" />
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
<xsd:element name="RestrictNetworkInterconnect" type="xsd:boolean"
minOccurs="0" />
<xsd:element ref="ShortCodes" minOccurs="0" />
</xsd:sequence>
<xsd:attribute name="GUID" type="xsd:string" />
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

3.10 INCOMINGGROUPIDS

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace=""
  xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="../common/ws_object.xsd" />

  <xsd:element name="IncomingGroupIds" substitutionGroup="Entity">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="IncomingGroupIdsList" type="xsd:string"/>
        <xsd:element name="PBXAddress" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

3.11 OUTGOINGGROUPIDS

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace=""
  xmlns="" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="../../../common/ws_object.xsd" />

  <xsd:element name="OutgoingGroupIds" substitutionGroup="Entity">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="OutgoingGroupIdsList" type="xsd:string"/>
        <xsd:element name="PBXAddress" type="xsd:string"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

4 Appendix B – Understanding the Data

4.1 ProvisionIPAddress

- **IPAddress** – IPAddress of the provisioned IP500 V2 system in the solution.

4.2 System

- **Name** – A name to identity the system.
- **MajorVersion** – Major version of the system
- **MinorVersion** – Minor version of the system
- **Configuration** – Operating mode of the system. The value represents the following:
 - A Law //0
 - JAPAN //1
 - MU Law //2
 - Partner //3
 - NOStar //4
 - Branch //5
 - Quick mode A law //6
 - Quick mode MU law //7
 - IP
- **BuildVersion**: Installed build version on the system
- **Telephony**
 - **Locale** – Telephony and Language settings.
 - **ModemTxLevel** -
 - **ModemCDThreshold**
 - **ModemNoActivityTimeout**

- Voicemail
 - Type
- LAN
 - ID
 - IPAddress
 - IPMask
 - PrimaryTranslationIPAddress
 - EnableNAT
 - RIPMode
 - dhcp_address_number
 - DHCPmode
 - NetworkTopologyDiscovery
 - StunServerAddress
 - ConnectivityType
 - RefreshTime
 - PublicIPAddress
 - PublicPortUdp
 - PublicPortTcp
 - PublicPortTls
 - STUNPort
 - RunOnFailure
 - IsBehindFirewall
 - VoipGatekeeper
 - H323GatekeeperEnable
 - H323RemoteExtnEnable
 - H323RemoteExtnCallSignalPort
 - H323SignalingOverTLS
 - SipTrunksEnable
 - SipRegistrarEnable
 - SipRemoteExtnEnable
 - DomainName
 - SipUdpEnable
 - UdpPort
 - RemoteUdpPort
 - SipTcpEnable
 - TcpPort
 - RemoteTcpPort
 - SipTlsEnable
 - TlsPort
 - RemoteTlsPort
 - ChallengeExpiryTime
 - RtpPortMin
 - RtpPortMax
 - RtpPortMinNat
 - RtpPortMaxNat
 - RtcpMonitorEnable

- AutoCreateExtn
- AutoCreateUser
- RemoteCallSignallingPort
- DHCPPrimarySSON
- DHCPSecondarySSON
- DHCPVlan
- DHCPVlanSSON
- DHCPVlanIDs
- RTCPCollectorIpAddress
- SIPRegistrarFQDN
- SipRegistrarAutoCreateExtnOrUser
- LanDHCPPools
 - ApplyToAvayaIPPhonesOnly
 - DHCP Pools
 - DHCPStartAddress
 - DHCPSubnetMask
 - DHCPDefaultRouter
 - DHCPPoolSize
- Scope
- InitialKeepalives
- PeriodicTimeout
- DiffServSettings
 - DSCP
 - VideoDSCP
 - DSCPMask
 - SigDSCP
- FirewallProfile
- PBXLicense
 - EnableRemoteServer
 - EnableProxyViaPrimaryIPOLine
 - LicenseServerIPAddress
 - LicenseServerURL
 - LicenseServerURLSecondary
 - LicenseServerURN
 - LicenseServerPort
 - LicenseServerID
 - LicenseMode
 - LicenseGracePeriod
 - ReservedLicenses
 - VMProRecordingsAdmins
 - WaveUsers
 - VMProTtsPro
 - VMProTtsWindows
 - VMProTts3rdParty
 - CtiLinkPro

- EmbeddedVMPorts
- VMProPorts
- SipSessions
- ACCS
- BranchSystem
- SMTrunks
- UMSWebServices
- MacSoftphones
- ThirdPartyRecorder
- ServerEdition
- AvayaIPEndpoints
- ThirdPartyIPEndpoints
- Receptionist
- BasicUser
- OfficeWorker
- Teleworker
- MobileUser
- PowerUser
- AvayaSoftphone
- WebCollaboration
- UniversalPRIAdditionalchannels
- IPSecTunneling
- CentralizedEndpoints
- EssentialEdition
- PreferredEdition
- LicensedVersion
- PLDSHostId
- PLDSFileStatus
- SelectLicensingStatus
- GraceDongleSerialNumber
- DongleSerialGraceDays
- GracePLDSHostId
- LicensingBasicUser

4.3 Licenses

- **GUID** – Hex Digit String
 - Globally Unique Identifier that is used to uniquely identify across the system the particular license
- **LicenseKey** – For ADI licenses it would be the actual license value. For other licenses it will be a license name.
- **Source** – Integer with value range of 0 to 4
 - Local=0 - The license is local to the IP Office (could be ADI or PLDS).

- WebLM=1 – The license is from a WebLM server.
- MappedLocal=2 – License is mapped from a local PLDS license.
- MappedRemote=3 – License is mapped from a remote PLDS license.
- MappedVirtual=4 – License is mapped from a virtual PLDS license.
- **Type** – Integer value indicating the type of license that the license key represents. The integer value represents a license that is dependent on the Mode. In other words there are different mappings of the licenses depending on the mode.
 - If the Mode is ADI (i.e. 1) the translation list is as follows:
 - "Invalid", // 0
 - "Unused (1)", // 1
 - "CTI Link Pro", // 2
 - "Wave User", // 3
 - "Integrated Messaging", // 4
 - "Preferred Edition", // 5
 - "Microsoft CRM Integration (users)", // 6
 - "CCC Spectrum Wallboards", // 7
 - "DECT Integration (ports)", // 8
 - "Phone Manager Pro", // 9
 - "Phone Manager Pro IP Audio Enabled (users)", // 10
 - "Compact Business Centre", // 11
 - "CCC Server", // 12
 - "CCC Agents", // 13
 - "CCC PC Wallboards", // 14
 - "CCC Supervisors", // 15
 - "Receptionist AddOn", // 16
 - "eBLF", // 17
 - "Additional Voicemail Pro (ports)", // 18
 - "CCC Designer (users)", // 19
 - "CCC Agent Rostering", // 20
 - "IP End-points", // 21
 - "Phone Manager Pro (per seat)", // 22
 - "CCC EMail", // 23
 - "CCC Chat", // 24
 - "AUDIX Voicemail", // 25
 - "VMPro Networked Messaging", // 26
 - "VMPro Database Interface", // 27
 - "VMPro VB Script", // 28
 - "VMPro Recordings Administrators", // 29
 - "VMPro Outlook Interface", // 30
 - "VMPro TTS (Scansoft)", // 31
 - "VMPro TTS (Generic)", // 32
 - "Conferencing Center", // 33
 - "Small Office Edition VCM (channels)", // 34
 - "Small Office Edition WiFi", // 35

- "IPSec Tunnelling", // 36
- "Proactive Reporting", // 37
- "Report Viewer", // 38
- "Mobile Twinning (users)", // 39
- "Advanced Small Community Networking", // 40
- "IP500 Standard Networking (Add 4 channels)", // 41
- "IP500 Upgrade Express to Professional", // 42
- "IP500 Standard Networking (Base 4 channels)", // 43
- "IP500 VCM Channels", // 44
- "SIP Trunk Channels", // 45
- "VPN IP Extensions", // 46
- "IP500 Universal PRI (Additional channels)", // 47
- "RAS LRQ Support (Rapid Response)", // 48
- "Premium Product Support - IP Office Standard Edition", // 49
- "Premium Product Support - IP Office Professional Edition", // 50
- "Basic Product Support - IP Office Standard Edition", // 51
- "Basic Product Support - IP Office Professional Edition", // 52
- "UMS Web Services", // 53
- "CCR Supervisor", // 54
- "Customer Service Agent", // 55
- "CCR Designer", // 56
- "CCR CCC Upgrade", // 57
- "IP16xx Terminals", // 58
- "Third Party API", // 59
- "Software Upgrade", // 60
- "one-X Portal for IP Office", // 61
- "Avaya IP Endpoints", // 62
- "Supervisor User", // 63
- "Embedded Voicemail Channels", // 64
- "Teleworker User", // 65
- "Mobile User", // 66
- "Power User", // 67
- "Advanced Edition", // 68
- "Office Worker User", // 69
- "Small Site Software Upgrade", // 70
- "Branch Edition", // 71
- "Essential to Branch Edition Migration", // 72
- "Preferred/Advanced to Branch Edition Migration", // 73
- "Centralized Endpoints", // 74
- "IP Endpoint to Enterprise Branch User Migration", // 75
- "Power User for Linux", // 76
- "Preferred Edition (VoiceMail Pro for Linux)", // 77
- "Avaya IP Endpoints for Linux", // 78

- "Receptionist for Linux", // 79
- "SIP Trunk Channels for Linux", // 80
- "Preferred Edition Additional VoiceMail Ports for Linux", //
- 81
- "3rd Party IP Endpoints for Linux", // 82
- "Third Party API for Linux", // 83
- "Voice Networking Channels for Linux", // 84
- "Mobile Worker for Linux", // 85
- "Essential Edition", // 86
- "Mobile User Upgrade", // 87
- "Office Worker User Upgrade", // 88
- "UMS Web Services for Linux", // 89
- "Branding", // 90
- "VMPro TTS Professional", // 91
- "R8+ Preferred Edition", // 92
- "CCR SUP Linux", // 93
- "Customer Service Supervisor Linux", // 94
- "Customer Service Agent Linux", // 95
- "Advanced Edition Linux", // 96
- "Server Edition", // 97
- "Server Edition Upgrade", // 98
- "Server Edition R9", // 99
- "Avaya Softphone", // 100
- "Allow Virtualization", // 101
- "Server Edition for Russia", // 102
- "Software Upgrade for Russia", // 103
- "Outbound Contact Express", // 104
- "SCN for CTI Link Pro", // 105
- "Server Edition Upgrade 5", // 106
- "Server Edition Upgrade 10", // 107
- "Web Collaboration", // 108
- "SM Trunk Channels", // 109
- "Server Edition R9.1", // 110
- "Server Edition for Russia R9.1", // 111
- "Branch System", // 112
- "Avaya Legacy Softphone", // 113
- "Avaya Mac Softphone", // 114
- "Avaya Contact Center Select", // 115
- "IP Office Select", // 116
- "WebLM Model 9.1", // 117
- "Devlink3 External Recorder", // 118
- "WebLM Model", // 119
- "Basic User", // 120
- "Media Encryption Disable", // 121
- "Call Analytics", // 122
- "Call Analytics SE", // 123

- "Server Edition R10" // 124
- If the is Mode PLDS (i.e. 2) the translation list is as follows:
 - "Invalid", // 0
 - "WebLM", // 1
 - "CTI Link Pro", // 2
 - "VoiceMail Ports", // 3
 - "VoiceMail Recordings Administrator", // 4
 - "Embedded Voicemail Channels", // 5
 - "Teleworker", // 6
 - "Mobile Worker", // 7
 - "Office Worker", // 8
 - "Softphone Lite", // 9
 - "VoiceMail TTS Pro", // 10
 - "IP Sec", // 11
 - "Power User", // 12
 - "CCR Agent", // 13
 - "CCR Supervisor", // 14
 - "Avaya IP Endpoints", // 15
 - "IP Trunk Channels", // 16
 - "SIP Trunk Channels", // 17
 - "PRI Channels", // 18
 - "TAPI CC", // 19
 - "WAV", // 20
 - "IP End-points", // 21
 - "Centrex End-points", // 22
 - "Essential Edition", // 23
 - "Advanced Edition", // 24
 - "Preferred Edition", // 25
 - "Server Edition", // 26
 - "Branding", // 27
 - "Web Collaboration", // 28
 - "SM Trunk Channels", // 29
 - "Software Upgrade", // 30
 - "Software Small Upgrade", // 31
 - "Russia Upgrade", // 32
 - "WebLM 9.1", // 33
 - "MAC Softphone", // 34
 - "Russia Server Edition", // 35
 - "Branch", // 36
 - "ACCS", // 37
 - "TTS Scansoft", // 38
 - "TTS Generic", // 39
 - "VoiceMail UMS", // 40
 - "Select", // 41
 - "Server Edition Upgrade", // 42
 - "Virtualization", // 43

- "Legacy Softphone" // 44
- **Status** – Integer value indicating the current status of the license
 - Invalid=0 - The license is not valid for this system.
 - Valid=1 – The license is valid and usable.
 - Expired=2 – The license is valid but has expired.
 - TokenExpired=3
 - Unknown=4 – The license status is unknown.
 - Dormant=5
 - Obsolete=6 – The license is obsolete for this system.
- **Quantity** – Integer value indicating the total number of this license that are available.
- **FreeInstances** – Integer value indicating the number of free instances of this license that are currently available. If a remote license server is being used, this number may be the same as the Quantity value provided.
- **ExpiryDate** – The expiry date for this license. Typical format is YYYY/MM/DD
- **Mode** – Integer value the mode of the license
 - Unknown = 0
 - ADI = 1
 - PLDS = 2 (Note: If it is a WebLM license, the mode is PLDS).
 - Virtual = 3
- **DisplayName** – Readable display name of the License.
- **AvailableUserPackageLicenseOfType** –
- **IsGrace** – Boolean indicating if the license has expired and we are in a grace period or not.
 - False – Not in grace period (i.e. license is still valid and not expired).
 - True – License is in grace period and may not be available in some future.
- **PBXAddress** – IP Address of the IP Office using the license.
- **Last-Modified** – Day of Week, Date, time and timezone when this license key was last modified. Example 'Mon, 29 Feb 2016 16:41:31 GMT'
- **Etag** – Entity Tag. Typically the date/time that the license was created/obtained/changed.

4.4 Users

- **GUID** – Hex Digit String
 - Globally Unique Identifier that is used to uniquely identify across the system the particular user.
- **AssignedPackage** – Integer
 - Indicates the User Profile which controls the features that they can be configured with.

- Basic – 1
- Teleworker – 2
- Mobile - 3
- Power – 4
- OfficeWorker – 5
- BranchUser – 6
- CentralizedUser – 7
- NonLicensedUser - 8
- **DNDExceptions**
 - List ('|' separated) of telephone numbers which are exceptions to the Do Not Disturb setting. Both internal extensions and external telephone numbers can be used. A range of numbers can be used by making use of the wildcards "N" and "X" in the number.
- **DoNotDisturb** – Boolean (true/false)
 - If 'true', the user's extension is considered busy except for calls coming from sources listed in their DNDExceptions list.
- **ExpansionType** – Integer
 - Integer value indicating the type of phone expansion on the user device.
 - ExtNPeripheralNone (None) – 0
 - ExtNPeripheralEU24x1 (46xx button module) – 1
 - ExtNPeripheralEU24x2 (46xx button module) – 2
 - ExtNPeripheralXM24x1 (64xx button module) – 3
 - ExtNPeripheralXM24x2 (64xx button module) – 4
 - ExtNPeripheralDSS4450x1 (44xx button module) – 5
 - ExtNPeripheralDSS4450x2 (44xx button module) – 6
 - ExtNPeripheralSBM32x1 (Sage button module) – 7
 - ExtNPeripheralSBM32x2 (Sage button module) – 8
 - ExtNPeripheralSBM32x3 (Sage button module) – 9
 - ExtNPeripheralBM24x1 (Spice button module) – 10
 - ExtNPeripheralBM24x2 (Spice button module) – 11
 - ExtNPeripheralBM24x3 (Spice button module) – 12
 - ExtNPeripheralBM12x1 (94xx/95xx button module) – 13
 - ExtNPeripheralBM12x2 (94xx/95xx button module) – 14
 - ExtNPeripheralSBM12x3 (94xx/95xx button module) – 15
 - ExtNPeripheralBstT7316Ex1 (T7316E KIM module) – 16
 - ExtNPeripheralBstT7316Ex2 (T7316E KIM module) – 17
 - ExtNPeripheralBstT7316Ex3 (T7316E KIM module) – 18
 - ExtNPeripheralBstT7316Ex4 (T7316E KIM module) – 19
 - ExtNPeripheralBstT7316Ex5 (T7316E KIM module) – 20
 - ExtNPeripheralBstT7316Ex6 (T7316E KIM module) – 21
 - ExtNPeripheralBstT7316Ex7 (T7316E KIM module) – 22
 - ExtNPeripheralBstT7316Ex8 (T7316E KIM module) – 23
 - ExtNPeripheralBstT7316Ex9 (T7316E KIM module) – 24
 - ExtNPeripheralBstM7324x1 (M7324 KLM module) – 25

- ExtnPeripheralBstM7324x2 (M7324 KLM module) – 26
- ExtnPeripheralSIPLCD18x1(11xx SIP button module) - 27
- ExtnPeripheralSIPLCD18x2(11xx SIP button module) – 28
- ExtnPeripheralSIPLCD18x3(11xx SIP button module) – 29
- ExtnPeripheralSIPLCD12x1(12xx SIP button module) - 30
- ExtnPeripheralSIPLCD12x2(12xx SIP button module) - 31
- ExtnPeripheralSIPLCD12x3(12xx SIP button module) - 32
- ExtnPeripheralSIPLCD12x4(12xx SIP button module) - 33
- ExtnPeripheralSIPLCD12x5(12xx SIP button module) - 34
- ExtnPeripheralSIPLCD12x6(12xx SIP button module) - 35
- ExtnPeripheralSIPLCD12x7(12xx SIP button module) – 36
- ExtnPeripheralSIPLCD18x1(12xx SIP button module) - 37
- ExtnPeripheralSIPLCD18x2(12xx SIP button module) - 38
- **Extension** – Digit String
 - Extension number for this user.
- **FlareEnabled** – Boolean (true/false)
 - 'true' indicates the user may use Avaya Communicator.
- **ForceAccountCode** – Boolean (true/false)
 - 'true' indicates the user must enter a valid account code when making any external call.
- **FullName** – String
 - String used for holding the user's full name. Recommended format is <first name><space><last name> in order to be used correctly by the Voicemail Dial By Name features. Name should not start with a space. The punctuation characters '#', '?', '/', '^', '>', '<comma>' and '.' are not allowed.
- **IdleLinePreference** – Boolean (true/false)
 - 'true' indicates that line selection will be done based on the first available appearance button.
- **LoginCode** – Digit String
 - 4 to 31 digits used as part of a login sequence in order to allow a user to make use of an extension as if it was their own phone.
 - IP phone users should not have more than 13 digits set. It is used during IP phone registration.
 - Code can be used for Hot Desking.
 - Users can only log out if they have a login code set.
 - Supports the short code feature 'Change Login Code'
 - If a user has a login code set, it is used by the 'Outgoing Call Bar Off' short code feature.
 - If a user has a login code set, access to a range of programmable button features will require entry of the login code.
- **MobilityFeatures** – Boolean (true/false)
 - 'true' indicates that the user may use the Mobile VoIP Client as the IP Office Phone. User must be a 'Power User'.

- **Name** - String
 - String up to 15 characters in length containing the User's account name. Name should not start with a space. The punctuation characters '#', '?', '/', '^', '>', '<comma>' and '.' are not allowed.
- **OutgoingCallBar** – Boolean (true/false)
 - 'true' will stop the user from making any external calls except for emergency calls.
- **Password** – String
 - Up to 31 alphanumeric characters. Used by a number of applications such as One-X, Soft Console and TAPI. Note that this is not the voicemail mailbox password.
 - Password must follow the Password Complexity rules. If not met, the configuration will still be saved but an error will be raised.
- **PhoneType** – Integer
 - An integer value indicating the type of phone connected to this user and extension. It is possible that the phone type is unknown, especially if the user has not yet ever logged in.
 - There are too many phone types to list here so a list of the common ones follow:
 - Unknown Type – 46
 - Unknown Digital – 47
 - Unknown VoIP – 48
 - Unknown SIP – 50
 - Unknown Device – 51
 - Avaya 9620 – 53
 - Avaya 9630 – 54
 - Avaya 9610 – 60
 - Avaya 9640 – 61
 - Avaya 9650 – 62
 - Avaya 1603 – 63
 - Avaya 1608 – 64
 - Avaya 1616 – 65
 - Avaya 1703 – 66
 - Avaya 1708 – 67
 - Avaya 1716 – 68
 - Avaya 1403 – 79
 - Avaya 1408 – 80
 - Avaya 1416 – 81
 - Avaya 1450 – 82
 - Avaya 1603L – 89
 - Avaya 1608L – 90
 - Avaya 1616L – 91
 - Avaya 9620Sip – 100
 - Avaya 9630Sip – 101
 - Avaya 9640Sip – 102

- Avaya 9650Sip – 103
- Avaya 96xxSip – 104
- Avaya 1603Sip – 105
- Avaya 9611 – 111
- Avaya 9621 – 112
- Avaya 9641 – 113
- Avaya 1010Sip – 116
- Avaya 1040Sip – 117
- Avaya 1120ESip – 118
- Avaya 1140ESip – 119
- Avaya 1220Sip – 120
- Avaya 1230Sip – 121
- Avaya 1692Sip – 122
- nortelT7316E – 128
- nortelM7310 – 129
- nortelM7310BLF – 130
- nortelT7316 – 131
- nortelM7324 – 132
- nortelDoorphone – 133
- nortelT7208 – 134
- nortelM7208 – 135
- nortel T7100- 136
- nortelM7100 -137
- nortelT7000 – 138
- nortelM7310N – 142
- nortelM7100N – 144
- nortelM7324N – 145
- nortelM7208N – 146
- AvayaB179Sip – 147
- Avaya One-X – 150
- Avaya Communicator – 151
- Avaya D160 – 152
- Unknown SIP DECT - 153
- **Receptionist** – Boolean (true/false)
 - 'true' indicates that the user may use the SoftConsole application.
- **RemoteWorker** – Boolean (true/false)
 - 'true' indicates that the user is allowed to use an H.323 or SIP remote extension. AssignedPackage must be Teleworker or Power User.
- **SIPContact** – String
 - String that is used when the Contact field of the SIP URI being used for a SIP call is set to 'Use Internal Data'
- **SIPName** – String
 - String that is used when the 'From' field of the SIP URI being used for a SIP call is set to 'Use Internal Data'.

- **SpecificBstType** – Integer
 - Value indicating the specific type of Business Series Terminal (i.e. BST) if a BST set is being used by the user.
 - Not a BST device – (-1)
 - ExtnDeviceBstM7310 – 129
 - ExtnDeviceBstM7208 – 134
 -
- **TwinningType** – Integer
 - The type of twinning set for the user
 - No Twinning – 0
 - Internal Twinning – 1
 - Mobile Twinning - 2
- **UMSWebServices** – Boolean (true/false)
 - 'true' indicates that the user may use any of the Voicemail Pro UMS services to access their voicemail messages(IMAP email client, web browser or Exchange mailbox).
- **VoicemailCode** – Digit String
 - 0 to 15 digits. A code that a user would use to access their voicemail mailbox. Codes that match a mailbox extension, repeat the same number (i.e. 222222) or are a sequence of numbers (i.e. 1234) are not allowed.
- **VoicemailEmail** – String
 - A blank indicates no voicemail email features.
 - Email address of where the Voicemail system will send voicemails.
- **VoicemailOn** – Boolean (true/false)
 - 'true' indicates that the mailbox is used by the system to answer the user's unanswered calls or those when the user is busy.
 - 'false' does not disable the use of the user's mailbox. Messages can still be forwarded to the mailbox and recordings can be place in it.
- **WebCollaboration** – Boolean (true/false)
 - 'true' indicates that the user has web collaboration capabilities.
 - 'false' indicates that the user does NOT have web collaboration capabilities.
- **Xdirectory** – Boolean (true/false)
 - 'true' indicates that the user is excluded from the system directory list.

4.5 Extensions

- **GUID** – Hex Digit String
 - Globally Unique Identifier that is used to uniquely identify across the system the particular extension.
- **Id** – Integer

- Integer value representing the physical ID of the extension port.
This value is allocated by the system and is not configurable.
- **Extension** – Digit String
 - Digit string representing the unique extension number of the device.
- **TypeInfo** – Integer
 - Value representing the type of extension
 - Normal – 0
 - H.323 – 1
 - ADMM (Avaya DECT Mobility) – 2
 - SIP – 3
 - D100 - 4
- **CallerDisplayType** – Integer
 - Value representing how the caller display information is presented for analog extensions. For digital and IP extensions, the value is fixed to On.
 - Off – 0
 - On – 1
 - UK – 2
 - UK20 – 3
 - DTMFA – 4
 - DTMFB – 5
 - DTMFC – 6
 - FSKA – 7
 - FSKB – 8
 - FSKC – 9
 - FSKD – 10
 - DTMFD – 11
 - DTMFE – 12
 - DTMFF – 13
 - DTMFG – 14
 - DTMFH – 15
 - DTMFI – 16
 - DTMFJ – 17
 - FSKE – 18
 - FSKF – 19
 - FSKG – 20
 - FSKH – 21
 - FSKI – 22
 - FSKJ - 23
- **Module** – Integer
 - Indicates external expansion module on which the port for the extension is located.
- **Port** – Integer
 - Port number on the above module. VoIP extensions always report as 0.

- **Location** – Integer
 - The physical location number that the extension is associated with. This location allows emergency services to identify the source of an emergency call.

4.6 HuntGroups

- **GUID** – Hex Digit String
 - Globally Unique Identifier that is used to uniquely identify across the system the particular Hunt Group.
- **Name** – String
 - Hunt Group name up to 15 characters. Name must be unique and is case sensitive. Should not start with a space and should not use the punctuation characters '#', '?', '/', '^', '>', '<comma>' and '.'
- **Extension** – Digit String
 - 2 to 15 digits which represents the extension number of the hunt group. Extension numbers in the range 8897 to 9999 are reserved and cannot be used.
- **GroupRingMode** – Integer
 - Value indicating how the system determines the hunt group member to ring first.
 - Collective - 0
 - Sequential - 1
 - Rotary - 2
 - Longest Waiting - 3
 - Collective Call Waiting – 4
- Collective → All available phones in the Member list ring simultaneously.
- Sequential → Each extension is run in order, one after the other, starting from the first extension in the list.
- Rotary → Each extension is rung in order, one after the other, starting from the extension after the last extension used.
- Longest Waiting → The extension that has been unused the longest period rings first, followed by the extension that has been idle second longest and so on.
- Collective Call Waiting → Same as collective but with hunt group call waiting also enabled.
- **MembersList** – String
 - List ('|' separated) of extensions representing the hunt group members.
- **ServiceMode** – Integer
 - Mode of service that the hunt group is in.
 - Out of Service – 0

- In Service – 1
- Night Service – 2
- Out of Service → Calls are redirected using the Out of Service Fallback Group setting.
- In Service → The hunt group is enabled.
- Night Service -> Calls are redirected using the Night Service Fallback Group setting.
- **VoicemailOn** – Boolean
 - 'true' indicates that the system will use the mailbox to answer any calls to the group that reach the Voicemail Answer Time.
 - 'false' does not disable use of the hunt group mailbox. Messages can still be forward to the mailbox and recordings can be placed in it. The mailbox can also still be accessed to collect messages.
- **IsGroupDistributed** – Boolean
 - 'true' indicates that some members in the list are located on other systems within the network.

4.7 ShortCodes

- **GUID** – Hex Digit String
 - Globally Unique Identifier that is used to uniquely identify across the system the particular short code.
- **Code** – String
 - The dialing digits used to trigger the short code. This has a maximum length of 31 characters.
- **TelephoneNumber** – String
 - The number dialed by the short code or parameters for the short code feature. It can contain numbers and characters.
- **LineGroupId** – Integer
 - Integer value representing the Outgoing Group Id.
- **Feature** – Integer
 - Value representing the action to be performed. A long list of features is available of which there exist too many to list all of them here. Below are some of the more common ones.
 - Forward Unconditional On – 0
 - Forward Unconditional Off – 1
 - Forward on Busy On – 2
 - Forward on Busy Off – 3
 - Forward on No Answer On – 4
 - Forward on No Answer Off – 5
 - Forward Number – 6
 - Do Not Disturb On – 7
 - Do Not Disturb Off – 8

- VoicemailCollect – 16
- Call Pickup Extension – 40
- Extension Login – 46
- Extension Logout – 47
- Park Call – 48
- UnPark Call – 49
- Conference Add – 62
- Dial Extension – 69
- Dial Emergency – 78
- Blind Transfer – 140
- Mobile Twinning On – 166
- Mobile Twinning Off – 167
- Mobile Twinning Pickup – 168
- Stamp Log - 216
- **ForceAccountCode** – Boolean
 - 'true' indicates that for short codes which result in dialing of a number, the user will be prompted to enter a valid account code before the call is allowed to continue.

4.8 Incoming Call Routes

- **GUID** – Hex Digit String
 - Globally Unique Identifier that is used to uniquely identify across the system the particular Incoming Call Route.
- **LineGroupId** – Integer
 - Value representing the trunk's incoming line group receiving the call.
- **CLI** – String
 - Number to match the caller's ICLID provided with the call. Field is matched left to right. Options are:
 - Full Telephone Number
 - Partial telephone number (i.e. like just the area code)
 - '!' – matches calls where the ICLID was withheld.
 - '?' – number unavailable
 - Blank – for all.
- **Destination** – String
 - Location to route the call when a match is found. Destination could be an extension, voicemail, hunt group name, Auto Attendant,
- **IncomingNumber** – String
 - The incoming number on the trunk/line to match against. Matching is done on a right to left manner. Options are:
 - '*' – Incoming CLI matching takes precedence.

- ‘-’ – Left to right exact length matching.
- ‘X’ – Single digit wildcard.
- ‘i’ – ISDN Calling Party Number National.

4.9 Incoming Line Group Ids

- **IncomingGroupIdsList** – String
 - List (‘|’ separated) of incoming group ids defined in the system.
- **PBXAddress** – String
 - LAN 1 IP Address of the System to which the Incoming Group Ids applies.

4.10 Outgoing Line Group Ids

- **OutgoingGroupIdsList** – String
 - List (‘|’ separated) of outgoing group ids defined in the system.
- **PBXAddress** – String
 - LAN 1 IP Address of the System to which the Outgoing Group Ids applies.

5 Appendix C – Common code snippets

5.1 Sample code for GET method

JavaScript

```
var xhr = new XMLHttpRequest();
xhr.open("GET", URI, true);
xhr.setRequestHeader("X-User-Agent", "Avaya-SDKUser");
xhr.setRequestHeader("X-User-Client", "Avaya-WebAdmin");
xhr.setRequestHeader("Content-Type", "application/xml");
xhr.setRequestHeader("Authorization", "Basic " + btoa("username:password"));

xhr.send();
```

Java

```
try
{
    URL url = new URL(URI);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setRequestMethod("GET");
    conn.setRequestProperty("X-User-Agent", "Avaya-SDKUser");
    conn.setRequestProperty("X-User-Client", "Avaya-WebAdmin");
    conn.setRequestProperty("Content-Type", "application/xml");
    conn.setRequestProperty("Cookie", cookies);

    String authStr = username+":"+password;
    byte[] bytesEncoded = Base64.encodeBase64(authStr.getBytes());
    String authEncoded = new String(bytesEncoded);
    conn.setRequestProperty("Authorization", "Basic "+authEncoded);

    if(conn.getResponseCode() != 200) {
        throw new RuntimeException("Failed : HTTP error code : "
            + conn.getResponseCode());
    }

    //getCookies(); Call this function during the authenticate web service

    BufferedReader br = new BufferedReader(new
    InputStreamReader((conn.getInputStream())));

    String output;
    while ((output = br.readLine()) != null) {
        System.out.println(output);
    }
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
```

.NET

```
CookieContainer cookieContainer = new CookieContainer();
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(uri);
request.Credentials = CredentialCache.DefaultCredentials;
request.Method = "GET";
request.ContentType = "application/json";
request.Headers.Add("X-User-Client", "Avaya-WebAdmin");
```

```
request.Headers.Add("X-User-Agent", "Avaya-SDKUser");
String credentials =
Convert.ToBase64String(ASCIIEncoding.ASCII.GetBytes(username + ":" +
password));
request.Headers.Add("Authorization", "Basic " + credentials);
request.CookieContainer = cookieContainer;

HttpWebResponse response = (HttpWebResponse)request.GetResponse();
var resStream = response.GetResponseStream();
var reader = new StreamReader(resStream);
String responseContent = reader.ReadToEnd();

response.Close();
```

5.2 Sample code for PUT method

JavaScript

```
var xhr = new XMLHttpRequest();
xhr.open("PUT", URI, false);
xhr.setRequestHeader("X-User-Agent", "Avaya-SDKUser");
xhr.setRequestHeader("X-User-Client", "Avaya-WebAdmin");
xhr.setRequestHeader("Content-Type", "application/xml");
xhr.setRequestHeader("Authorization", "Basic " + btoa("username:password"));

xhr.send(putData);
```

Java

```
try
{
    URL url = new URL(URI);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setDoOutput(true);
    conn.setRequestMethod("PUT");
    conn.setRequestProperty("X-User-Agent", "Avaya-SDKUser");
    conn.setRequestProperty("X-User-Client", "Avaya-WebAdmin");
    conn.setRequestProperty("Content-Type", "application/xml");
    conn.setRequestProperty("Cookie", cookies);
    String authStr = username+":"+password;
    byte[] bytesEncoded = Base64.encodeBase64(authStr.getBytes());
    String authEncoded = new String(bytesEncoded);
    conn.setRequestProperty("Authorization", "Basic "+authEncoded);

    if(conn.getResponseCode() != 200) {
        throw new RuntimeException("Failed : HTTP error code : "
            + conn.getResponseCode());
    }

    //getCookies(); Call this function during the authenticate web service

    BufferedReader br = new BufferedReader(new
    InputStreamReader((conn.getInputStream())));

    String output;
    while ((output = br.readLine()) != null) {
        System.out.println(output);
    }
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
```

```
}
    e.printStackTrace();
}
```

.NET

```
CookieContainer cookieContainer = new CookieContainer();
HttpRequest request = (HttpRequest)WebRequest.Create(uri);
request.Credentials = CredentialCache.DefaultCredentials;
request.Method = "PUT";
request.ContentType = "application/json";
request.Headers.Add("X-User-Client", "Avaya-WebAdmin");
request.Headers.Add("X-User-Agent", "Avaya-SDKUser");
String credentials =
Convert.ToBase64String(ASCIIEncoding.ASCII.GetBytes(username + ":" +
password));
request.Headers.Add("Authorization", "Basic " + credentials);
request.CookieContainer = cookieContainer;

Stream dataStream = request.GetRequestStream();
dataStream.Write(putData, 0, putData.Length);
dataStream.Close();

HttpWebResponse response = (HttpWebResponse)request.GetResponse();
var resStream = response.GetResponseStream();
var reader = new StreamReader(resStream);
String responseContent = reader.ReadToEnd();

response.Close();
```

5.3 Sample code for POST method

JavaScript

```
var xhr = new XMLHttpRequest();
xhr.open("POST", URI, false);
xhr.setRequestHeader("X-User-Agent", "Avaya-SDKUser");
xhr.setRequestHeader("X-User-Client", "Avaya-WebAdmin");
xhr.setRequestHeader("Content-Type", "application/xml");
xhr.setRequestHeader("Authorization", "Basic " + btoa("username:password"));

xhr.send(postData);
```

Java

```
try
{
    URL url = new URL(URI);
    HttpURLConnection conn = (HttpURLConnection) url.openConnection();
    conn.setRequestMethod("POST");
    conn.setDoOutput(true);
    conn.setRequestProperty("X-User-Agent", "Avaya-SDKUser");
    conn.setRequestProperty("X-User-Client", "Avaya-WebAdmin");
    conn.setRequestProperty("Content-Type", "application/xml");
    conn.setRequestProperty("Cookie", cookies);
    String authStr = username+":"+password;
    byte[] bytesEncoded = Base64.encodeBase64(authStr.getBytes());
    String authEncoded = new String(bytesEncoded);
    conn.setRequestProperty("Authorization", "Basic "+authEncoded);

    if(conn.getResponseCode() != 200) {
        throw new RuntimeException("Failed : HTTP error code : ")
    }
}
```

```

        + conn.getResponseCode());
    }

    BufferedReader br = new BufferedReader(new
InputStreamReader((conn.getInputStream())));

    String output;
    while ((output = br.readLine()) != null) {
        System.out.println(output);
    }
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

```

.NET

```

HttpRequest request = (HttpRequest)WebRequest.Create(uri);
request.Credentials = CredentialCache.DefaultCredentials;
request.Method = "POST";
request.ContentType = "application/json";
request.Headers.Add("X-User-Client", "Avaya-WebAdmin");
request.Headers.Add("X-User-Agent", "Avaya-SDKUser");
String credentials =
Convert.ToBase64String(ASCIIEncoding.ASCII.GetBytes(username + ":" +
password));
request.Headers.Add("Authorization", "Basic " + credentials);
request.CookieContainer = cookieContainer;

Stream dataStream = request.GetRequestStream();
dataStream.Write(postData, 0, postData.Length);
dataStream.Close();

HttpWebResponse response = (HttpWebResponse)request.GetResponse();
var resStream = response.GetResponseStream();
var reader = new StreamReader(resStream);
String responseContent = reader.ReadToEnd();

response.Close();

```

5.4 Sample code for DELETE method

JavaScript

```

var xhr = new XMLHttpRequest();
xhr.open("DELETE", URI, false);
xhr.setRequestHeader("X-User-Agent", "Avaya-SDKUser");
xhr.setRequestHeader("X-User-Client", "Avaya-WebAdmin");
xhr.setRequestHeader("Content-Type", "application/xml");
xhr.setRequestHeader("Authorization", "Basic " + btoa("username:password"));

xhr.send(putData);

```

Java

```

try
{
    URL url = new URL(URI);

```

```

URLConnection conn = (URLConnection) url.openConnection();
conn.setRequestMethod("DELETE");
conn.setDoOutput(true);
conn.setRequestProperty("X-User-Agent", "Avaya-SDKUser");
conn.setRequestProperty("X-User-Client", "Avaya-WebAdmin");
conn.setRequestProperty("Content-Type", "application/xml");
conn.setRequestProperty("Cookie", cookies);
String authStr = username+":"+password;
byte[] bytesEncoded = Base64.encodeBase64(authStr.getBytes());
String authEncoded = new String(bytesEncoded);
conn.setRequestProperty("Authorization", "Basic "+authEncoded);

if(conn.getResponseCode() != 200) {
    throw new RuntimeException("Failed : HTTP error code : "
        + conn.getResponseCode());
}

BufferedReader br = new BufferedReader(new
InputStreamReader((conn.getInputStream())));

String output;
while ((output = br.readLine()) != null) {
    System.out.println(output);
}
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}

```

.NET

```

HttpRequest request = (HttpRequest)WebRequest.Create(uri);
request.Credentials = CredentialCache.DefaultCredentials;
request.Method = "POST";
request.ContentType = "application/json";
request.Headers.Add("X-User-Client", "Avaya-WebAdmin");
request.Headers.Add("X-User-Agent", "Avaya-SDKUser");
String credentials =
Convert.ToBase64String(ASCIIEncoding.ASCII.GetBytes(username + ":" +
password));
request.Headers.Add("Authorization", "Basic " + credentials);
request.CookieContainer = cookieContainer;

HttpWebResponse response = (HttpWebResponse)request.GetResponse();
var resStream = response.GetResponseStream();
var reader = new StreamReader(resStream);
String responseContent = reader.ReadToEnd();

response.Close();

```

5.5 Certificate validation for .NET

```
private static bool CertificateValidationCallback(object sender,
System.Security.Cryptography.X509Certificates.X509Certificate certification,
System.Security.Cryptography.X509Certificates.X509Chain chain, SslPolicyErrors
sslPolicyErrors)
{
    bool retVal = false;
    bool sslCertError = false;
    try
    {
        if (((sslPolicyErrors & SslPolicyErrors.None) == SslPolicyErrors.None)
||
                ((sslPolicyErrors &
SslPolicyErrors.RemoteCertificateChainErrors) ==
SslPolicyErrors.RemoteCertificateChainErrors))
        {
            sslCertError = false;
        }
        DateTime dtEnd =
DateTime.Parse(certification.GetExpirationDateString());
        DateTime dtStart =
DateTime.Parse(certification.GetEffectiveDateString());

        if (dtStart > DateTime.Now || dtEnd < DateTime.Now)
        {
            retVal = false;
            return retVal;
        }

        if (sslCertError == false)
        {
            retVal = true;
        }
    }
    catch (Exception)
    {
        retVal = false;
    }

    return retVal;
}
```

```
public static String getCookies(HttpURLConnection conn)
{
    String cookies = new String();
    Map<String, List<String>> headers = conn.getHeaderFields();
    String cookie = null;
    for (String h : headers.keySet()) {
        if ((null != h) && h.equals("Set-Cookie")) {
            List<String> cookieList = headers.get(h);

            for (Iterator<String> iterator = cookieList.iterator();
iterator.hasNext();)
            {
                String object = (String)iterator.next();
                connCookies += object + ";";
            }
        }
    }
}
```

```
        break;  
    }  
}  
return cookies;  
}
```