



Avaya Call Reporting 4.4 API Documentation

© 2018-2019, Avaya Inc.
All Rights Reserved.

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 in varying mediums which may include product information, operating instructions and performance specifications that are generally made available to users of products. 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 or on the express behalf of 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.

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

Hosted Service

THE FOLLOWING APPLIES ONLY IF YOU PURCHASE AN AVAYA HOSTED SERVICE SUBSCRIPTION FROM AVAYA OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE). THE TERMS OF USE FOR HOSTED SERVICES ARE AVAILABLE ON THE AVAYA WEBSITE, <https://support.avaya.com/LICENSEINFO> UNDER THE LINK “Avaya Terms of Use for Hosted Services” OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, AND ARE APPLICABLE TO ANYONE WHO ACCESSES OR USES THE HOSTED SERVICE. BY ACCESSING OR USING THE HOSTED 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 HOSTED SERVICE OR AUTHORIZE ANYONE TO ACCESS OR USE THE HOSTED SERVICE.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, <https://support.avaya.com/LICENSEINFO>, 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 INC., 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 INC. OR THE APPLICABLE AVAYA AFFILIATE (“AVAYA”).

Avaya grants You a license within the scope of the license types described below, with the exception of Heritage Nortel Software, for which the scope of the license is detailed below. Where the order documentation does not expressly identify a license type, the applicable license will be a Designated System License as set forth below in the Designated System(s) License (DS) section as applicable. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the documentation or other materials available to You. “Software” means computer programs in object code, provided by Avaya or an Avaya Channel Partner, whether as stand-alone products, pre-installed on hardware products, and any upgrades, updates, patches, bug fixes, or modified versions thereto. “Designated Processor” means a single stand-alone computing device. “Server” means a set of Designated Processors that hosts (physically or virtually) a software application to be accessed by multiple users. “Instance” means a single copy of the Software executing at a particular time: (i) on one physical machine; or (ii) on one deployed software virtual machine (“VM”) or similar deployment.

License type(s)

Designated System(s) License (DS). End User may install and use each copy or an Instance of the Software only: 1) on a number of Designated Processors up to the number indicated in the order; or 2) up to the number of Instances of the Software as indicated in the order, Documentation, or as authorized by Avaya in writing. Avaya may require the Designated Processor(s) to be identified in the order by type, serial number, feature key, Instance, location or other specific designation, or to be provided by End User to Avaya through electronic means established by Avaya specifically for this purpose.

Concurrent User License (CU). End User may install and use the Software on multiple Designated Processors or one or more Servers, so long as only the licensed number of Units are accessing and using the Software at any given time. A “Unit” means the unit on which Avaya, at its sole discretion, bases the pricing of its licenses and can be, without limitation, an agent, port or user, an e-mail or voice mail account in the name of a person or corporate function (e.g., webmaster or helpdesk), or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software. Units may be linked to a specific, identified Server or an Instance of the Software.

Cluster License (CL). End User may install and use each copy or an Instance of the Software only up to the number of Clusters as indicated on the order with a default of one (1) Cluster if not stated. “Cluster” means a group of Servers and other resources that act as a single system.

Enterprise License (EN). End User may install and use each copy or an Instance of the Software only for enterprise-wide use of an unlimited number of Instances of the Software as indicated on the order or as authorized by Avaya in writing.

Named User License (NU). You may: (i) install and use each copy or Instance of the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use each copy or Instance of the Software on a Server so long as only authorized Named Users access and use the Software. "Named User", means a user or device that has been expressly authorized by Avaya to access and use the Software. At Avaya's sole discretion, a "Named User" may be, without limitation, designated by name, corporate function (e.g., webmaster or helpdesk), an e-mail or voice mail account in the name of a person or corporate function, or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software.

Shrinkwrap License (SR). You may install and use the Software in accordance with the terms and conditions of the applicable license agreements, such as "shrinkwrap" or "clickthrough" license accompanying or applicable to the Software ("Shrinkwrap License").

Heritage Nortel Software

"Heritage Nortel Software" means the software that was acquired by Avaya as part of its purchase of the Nortel Enterprise Solutions Business in December 2009. The Heritage Nortel Software is the software contained within the list of Heritage Nortel Products located at <https://support.avaya.com/LicenseInfo> under the link "Heritage Nortel Products" or such successor site as designated by Avaya. For Heritage Nortel Software, Avaya grants Customer a license to use Heritage Nortel Software provided hereunder solely to the extent of the authorized activation or authorized usage level, solely for the purpose specified in the Documentation, and solely as embedded in, for execution on, or for communication with Avaya equipment. Charges for Heritage Nortel Software may be based on extent of activation or use authorized as specified in an order or invoice.

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Hosted Service, or hardware provided by Avaya. All content on this site, the documentation, Hosted 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. Unless otherwise stated, 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 Hosted 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://support.avaya.com/Copyright> 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 only 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 MPEG LA, L.L.C. SEE [HTTP://WWW.MPEGLA.COM](http://www.mpegla.com).

Service Provider

THE FOLLOWING APPLIES TO AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS OR SERVICES. THE PRODUCT OR HOSTED 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 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 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 MPEG LA, L.L.C. SEE [HTTP://WWW.MPEGLA.COM](http://www.mpegla.com).

Compliance with Laws

You acknowledge and agree that it is Your responsibility 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 Center 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.

Table of Contents

5	1.0 Recording Data
5	Obtaining Audio Files
6	Creating External Listen Links
8	Obtaining Recording Data
10	2.0 Historical Call Data
14	Creating an API-Enabled Report
15	3.0 Recording Pause/Resume
15	Obtaining Agent Mapping
18	4.0 Realtime Data
18	Obtaining a List of API-Enabled Metrics
19	Obtaining a List of API-Enabled Metric Values
22	Creating a Realtime Metric
24	5.0 Recording Pause/Resume
25	6.0 Screen Pop Profile
26	7.0 WebService Integration
26	Web Service Integration Profile Configuration
27	8.0 Web Chat Javascript API
27	Current Queue Status Request
28	Subscribe to Queue Status
28	Set Custom Chat Parameters
29	Set Active Skill
29	Get Active Skill
29	Manually Start Chat
29	Manually Start Xima Invite
30	9.0 Web Chat Cloud Service Integration
30	Checking a Session
31	Starting a Session
33	Receiving Messages
34	Sending Messages
35	10.0 Glossary

1.0. Recording Data

After creating a Service User within the Avaya Call Reporting UI, you will be given an authentication key. Anytime a request is made to the API, an authentication key must be present.

Add the following header to your API request:

```
Authorization : Bearer {authentication_key}
```

By having the authorization key present, you will be able to access everything that the Service User is set up in Avaya Call Reporting to have access to.

USE CASES

- Programmatically download audio files to be imported into another system
- Programmatically generate links to play recorded calls outside of Avaya Call Reporting

1.1. Obtaining Audio Files

```
GET /rest/api/v1/recordings/{recording-key}/audio?format={format}
```

An endpoint to download a single recording.

ARGUMENTS

Recording Key	The recording key for the recording for which the audio file is requested.
Format (optional)	The audio file type returned. Supported audio types are spx (default) and wav.

RESPONSE

If a successful request is made, then the response will be a file containing the audio of the recording in the specified file type.

POTENTIAL ERRORS

Status Code	Failure
400 Request	An invalid {format} was provided.
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The authenticated user doesn't have Avaya Call Reporting access to the recording.
404 Not Found	An invalid recording key was provided.

GET /api/v1/recordings/{call-key}/audio?format={format}

An endpoint to download all of the recordings on a call.

ARGUMENTS

Call Key	The call key for the call for which the audio file is requested.
Format (optional)	The audio file type returned. Supported audio types are spx (default) and wav.

RESPONSE

If a successful request is made, then the response will be a file containing the audio of the call’s recordings in the specified file type.

POTENTIAL ERRORS

Status Code	Failure
400 Request	An invalid format was provided.
401 Unauthorized	The authorization header wasn’t valid.
403 Forbidden	The authenticated user doesn’t have Avaya Call Reporting access to the call.
404 Not Found	An invalid call key was provided.

1.2. Creating External Listen Links

POST /rest/api/v1/recordings/{recording-key}/external-listen-link

Obtain an external listen link for a specific recording.

This requires that System Settings -> Basic Settings -> Externally Accessible Address in Avaya Call Reporting be set with protocol (e.g. http://), domain (e.g., mywebsite.com or 10.0.0.101), and port (if applicable, e.g., :9080), for example: http://10.0.0.101:9080.

ARGUMENTS

Recording Key	The recording key for a recording.
---------------	------------------------------------

RESPONSE

If a successful request is made then a json result is returned. The external listen link that is generated will be valid forever (Integer.MAX_VALUE days).

Example Response

```
{
  "listenLink": "http://www.my-chronicall.com/extlisten/recording/
index.html?id\u003dc9ab20e0-9ad7-4eda-90be-cb401c39f981"
}
```

POTENTIAL ERRORS

Status Code	Failure
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The authenticated user doesn't have Avaya Call Reporting access to the recording.
404 Not Found	An invalid recording key was provided.

POST /rest/api/v1/calls/{call-key}/external-listen-link

Obtain an external listen link for a specific call.

This requires that System Settings -> Basic Settings -> Externally Accessible Address in Avaya Call Reporting be set with protocol (e.g., http://), domain (e.g., mywebsite.com or 10.0.0.101), and port (if applicable, e.g., :9080), for example: http://10.0.0.101:9080.

ARGUMENTS

Call Key	The call key for the call
----------	---------------------------

RESPONSE

If a successful request is made, then a json result is returned. The external listen link that is generated will be valid forever (Integer.MAX_VALUE days).

Example Response

```
{
  "listenLink": "http://www.my-chronicall.com/extlisten/recording/
index.html?id\u003dc9ab20e0-9ad7-4eda-90be-cb401c39f981"
}
```

POTENTIAL ERRORS

Status Code	Failure
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The authenticated user doesn't have Avaya Call Reporting access to the call.
404 Not Found	An invalid call key was provided.

1.3. Obtaining Recording Data

```
GET /rest/api/v1/calls/{call-key}/recordings
```

An endpoint to obtain all of the recording records for a given call.

RESPONSE

If a successful request is made, then a json list of basic recording record data will be returned.

Example Response

```
[
  {
    "recordingKey": "83849370-5753-4186-9fc5-9c069e615734-1568735345298",
    "startTime": 1568735345298,
    "duration": 48974,
    "poolId": "d25839a1-2916-4825-8a36-299a1119f732",
    "size": 42443,
    "eventId": 50165,
    "recordingSystemId": 1,
    "status": "SAVED"
  },
  {
    "recordingKey": "0a72f610-8615-4ac9-8f1d-b914f9f164dc-1568739643644",
    "startTime": 1568739643644,
    "duration": 59347,
    "poolId": "308b6f52-1f61-4c4f-9ee4-543f37f3937f",
    "size": 49743,
    "eventId": 54035,
    "recordingSystemId": 1,
    "status": "SAVED"
  }
]
```

ARGUMENTS

startTime long	The start of the time frame in milliseconds since epoch.
endTime long	The end of the time frame in milliseconds since epoch.

RESPONSE

If a successful request is made, then a json list of basic recording record data will be returned. The result can be an empty list. <<glossary:Recording>>s for which the user does not have access will not be returned.

Example Response

```
[
  {
    "recordingKey": "83849370-5753-4186-9fc5-9c069e615734-1568735345298",
    "startTime": 1568735345298,
    "duration": 48974,
    "poolId": "d25839a1-2916-4825-8a36-299a1119f732",
    "size": 42443,
    "eventId": 50165,
    "recordingSystemId": 1,
    "status": "SAVED"
  },
  {
    "recordingKey": "0a72f610-8615-4ac9-8f1d-b914f9f164dc-1568739643644",
    "startTime": 1568739643644,
    "duration": 59347,
    "poolId": "308b6f52-1f61-4c4f-9ee4-543f37f3937f",
    "size": 49743,
    "eventId": 54035,
    "recordingSystemId": 1,
    "status": "SAVED"
  }
]
```

POTENTIAL ERRORS

Status Code	Failure
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The authenticated user doesn't have Avaya Call Reporting access to the call.

2.0. Historical Call Data

After creating a Service User within the Avaya Call Reporting UI, you will be given an authentication key. Anytime a request is made to the API, an authentication key must be present.

Add the following header to your API request:

```
Authorization : Bearer {authentication_key}
```

By having the authorization key present, you will be able to access everything that the Service User is set up in Avaya Call Reporting to have access to.

USE CASES

- Periodically export historical call data into another application
- Programmatically generate report files in different formats to be consumed by another department or application
- Pull time card data from Avaya Call Reporting into a staffing or payroll system
- Pull call records into a billing system (example: hospitality)
- Pull account summary usage data into a billing system (example: law firm billing by account code)

```
POST /rest/api/v1/historic-data/{historicDataKey}
```

An endpoint to run a report. Requires a json request body.

ARGUMENTS

historicDataKey	The UUID of the report that you are trying to run. More on how to obtain this later.
-----------------	--

```
Example Request Body
{
  "format": "json",
  "parameters": [
    {
      "name": "Report Timeframe",
      "valueType": "REPORT_TIMEFRAME",
      "value": {
        "type": "REPORT_TIMEFRAME",
        "start": "2019-09-09T00:00:00.000-06:00",
        "end": "2019-09-10T23:59:59.000-06:00"
      }
    },
    {
      "name": "Rows (Agent)",
      "valueType": "PBX_USERS",
      "value": {
        "type": "PBX_USERS",
        "pbxUsers": [
          {
            "key": "Agent1(200)"
          }
        ]
      }
    }
  ]
}
```

RESPONSE

If a successful request is made, then a report will be returned in the format that was requested. The following is an example of a JSON response.

JSON

```
{
  "reportId": 48,
  "executionId": "3eb08a9a-0f05-44b8-a0b0-ba7338a1d2e5",
  "title": "Agent Talking Summary",
  "timeframe": {
    "type": "REPORT_TIMEFRAME",
    "start": "2019-09-09T00:00:00-06:00",
    "end": "2019-09-10T23:59:59-06:00"
  },
  "skin": "Avaya Default",
  "columns": [
    {
      "key": "0",
      "title": "Agent",
      "format": "STRING"
    },
    {
      "key": "1",
      "title": "Inbound Calls",
      "format": "NUMBER"
    },
    {
      "key": "2",
      "title": "Outbound Calls",
      "format": "NUMBER"
    },
    {
      "key": "3",
      "title": "Internal Calls",
      "format": "NUMBER"
    },
    {
      "key": "4",
      "title": "Total Calls",
      "format": "NUMBER"
    },
    {
      "key": "5",
      "title": "Total Talking",
      "format": "DURATION"
    },
    {
      "key": "6",
      "title": "Avg Talking",
      "format": "DURATION"
    }
  ]
}
```

JSON

```
"dataFlag": "INCLUDED",
"summaryItems": [
  {
    "title": "Agents",
    "value": 1,
    "formattedValue": "1",
    "format": "NUMBER"
  },
  {
    "title": "Total Inbound Calls",
    "value": 4,
    "formattedValue": "4",
    "format": "NUMBER"
  },
  {
    "title": "Total Outbound Calls",
    "value": 6,
    "formattedValue": "6",
    "format": "NUMBER"
  },
  {
    "title": "Total Internal Calls",
    "value": 4,
    "formattedValue": "4",
    "format": "NUMBER"
  },
  {
    "title": "Total Calls",
    "value": 14,
    "formattedValue": "14",
    "format": "NUMBER"
  },
  {
    "title": "Avg Total Calls",
    "value": 14.0,
    "formattedValue": "14.0",
    "format": "NUMBER"
  },
  {
    "title": "Total Talking",
    "value": 12673424,
    "formattedValue": "3:31:13",
    "format": "NUMBER"
  },
  {
    "title": "Avg Talking",
    "value": 1810489,
    "formattedValue": "0:30:10",
    "format": "NUMBER"
  }
],
```


JSON

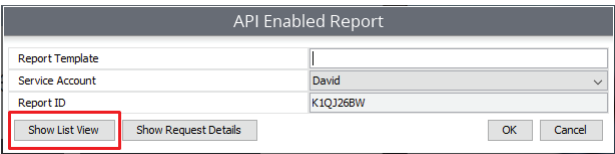
```
"rows": [
  {
    "values": {
      "0": {
        "value": "Agent1(200)"
      },
      "1": {
        "value": 4,
        "formattedValue": "4"
      },
      "2": {
        "value": 6,
        "formattedValue": "6"
      },
      "3": {
        "value": 4,
        "formattedValue": "4"
      },
      "4": {
        "value": 14,
        "formattedValue": "14"
      },
      "5": {
        "value": 12673424,
        "formattedValue": "3:31:13"
      },
      "6": {
        "value": 1810489,
        "formattedValue": "0:30:10"
      }
    }
  }
],
"charts": {
  "type": "CHARTS",
  "rows": []
}
```

POTENTIAL ERRORS

Status Code	Failure
400 Bad Request	Malformed JSON request body
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The authenticated user doesn't have Avaya Call Reporting access to view data for one or more of the requested agents or groups.

21. Creating an API-Enabled Report

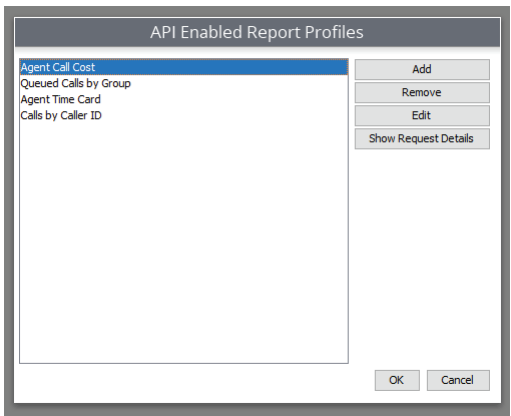
In order to make a request for a report, we'll need to first enable it for API use. Do so, navigate to the Avaya Call Reporting main page. On the left sidebar, select API Enablement > Historical Data Access. A pop up window will appear called "API Enabled Report." Click "Show List View," which is located in the bottom left corner. This will allow us to see all of the reports that are currently enabled for API use.



A new pop up window called "API Enabled Report Profiles" will appear. Select the report you would like to see request details for. If the report that you want to run does not exist, click the "Add" button, which is located in the top right corner of the window. Fill out the information that the "Add" window asks, which includes:

- **Report Template:** A searchable input that allows you to select any available report.
- **Service Account:** Select the service user that will have access to this report.
- **Report UUID:** The unique identifier for this report. This value will be used in the above URL as the historicDataKey.

Once the information has been entered, click "OK."



After you've added reports, you can view the "Request Details" for any report by clicking the "Show Request Details" button, which is located near the top right corner of the window. The Request Details view provides important information that is helpful when making API requests. When you click the button, you'll be prompted to select your report parameters, which will be used to generate details for your specific API request.

Click "OK" once all information has been entered. A new window will appear that provides you with all necessary information.

3.0. Recording Pause/Resume

After creating a Service User within the Avaya Call Reporting UI, you will be given an authentication key. Anytime a request is made to the API, an authentication key must be present.

Add the following header to your API request:

```
Authorization : Bearer {authentication_key}
```

By having the authorization key present, you will be able to access everything that the Service User is set up in Avaya Call Reporting to have access to.

USE CASES

Programmatically pause/unpause audio recording when an agent is accepting sensitive information (like a credit card number) over the phone.

3.1. Obtaining Agent Mapping

```
GET /rest/api/v1/agents/mapping
```

An endpoint to get the agent mappings.

RESPONSE

If a successful request is made, then a json result is returned.

```
Example Response

{
  "nthatcher": "Nate Thatcher(123)",
  ...
}
```

POTENTIAL ERRORS

Status Code	
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The authenticated user doesn't have Avaya Call Reporting access to the mapping.

RESPONSE

If a successful request is made then a json result is returned. Possible values for state are NO_RECORDING, RECORDING, and PAUSED.

Example Response

```
{
  "state": "RECORDING"
}
```

POTENTIAL ERRORS

Status Code	
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The authenticated user doesn't have Avaya Call Reporting access to the mapping.
404 Not Found	There is no agent found for the given AgentId.

POST /rest/api/v1/agents/{AgentId}/active-recording/pause

An endpoint to pause the active recording of an agent.

ARGUMENTS

AgentId	The AgentId of the desired agent.
---------	-----------------------------------

RESPONSE

If a successful request is made then a json result is returned. Possible values for state are NO_RECORDING, RECORDING, and PAUSED.

Example Response

```
{
  "state": "PAUSED"
}
```

POTENTIAL ERRORS

Status Code	
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The authenticated user doesn't have Avaya Call Reporting access to the mapping.
404 Not Found	There is no agent found for the AgentId.

POST /rest/api/v1/agents/{AgentId}/active-recording/resume

An endpoint to resume the active recording of an agent.

ARGUMENTS

AgentID	The AgentId of the desired agent.
---------	-----------------------------------

RESPONSE

If a successful request is made, then a json result is returned. This result indicates the new state for the active recording for the agent. Possible values for state are NO_RECORDING, RECORDING, and PAUSED.

Example Response

```
{
  "state": "RECORDING"
}
```

POTENTIAL ERRORS

Status Code	
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The authenticated user doesn't have Avaya Call Reporting access to the mapping.
404 Not Found	There is no agent found for the given AgentId.

4.0. Realtime Data

After creating a Service User within the Avaya Call Reporting UI, you will be given an authentication key. Anytime a request is made to the API, an authentication key must be present.

Add the following header to your API request:

```
Authorization : Bearer {authentication_key}
```

By having the authorization key present, you will be able to access everything that the Service User is set up in Avaya Call Reporting to have access to.

USE CASES

- Show Realtime metrics on a 3rd party dashboard with a periodic refresh
- Monitor Realtime metrics with a periodic refresh within a 3rd party web service
- Periodically refresh and show metrics like “Current Wait Time” and “Calls in Queue” on a customer’s web site

4.1. Obtaining a List of API-Enabled Metrics

```
GET /rest/api/v1/realtime/metrics
```

An endpoint to obtain a list of all API-enabled metrics.

RESPONSE

If a successful request is made, then a json result is returned.

RESPONSE BREAKDOWN

- "metricId": An ID to uniquely identify the metric. Not a UUID.
- "metricName": The name of the metric.
- "categoryType": The category type of the metric. Possible values are AGENT, GROUP, CALL_DIRECTION, ACCOUNT_CODE, and NONE.
- "categorySelection": A list of category values that have been selected for the metric.

Example Response

```
[
  {
    "metricId": "101",
    "metricName": "metricA",
    "categoryType": "AGENT",
    "categorySelection": [
      "Alice",
      "Bob",
      "Carol"
    ]
  },
  {
    "metricId": "102",
    "metricName": "metricB",
    "categoryType": "AGENT",
    "categorySelection": [
      "Dave",
      "Eve",
      "Frank"
    ]
  }
]
```

POTENTIAL ERRORS

Status Code	
401 Unauthorized	The authorization header wasn't valid.

4.2. Obtaining a List of API-Enabled Metric Values

```
GET /rest/api/v1/realtime/metrics/values?id={ids...}
```

An endpoint to obtain a list of values for specified API-enabled metrics. This returns values for metrics that are ready to return values ("results"), and provides a list of metrics not ready at the moment to return values ("initializing"). Values that are not initialized begin to initialize when requested so that they will be available for future requests. Values that are not requested for a long period of time will need to be reinitialized (automatically on the first request) before they are available again.

ARGUMENTS

IDs	Either one metric ID, or a comma-separated list of metric IDs.
-----	--

RESPONSE

If a successful request is made, then a json result is returned.

RESPONSE BREAKDOWN

- "results": Latest calculated values for the metrics. There are several different types of values that can be returned. See the **examples** of each type.
- "initializing": A list of metric IDs of metrics that are not yet initialized. For metrics with category selection, this will be a combination of the metric ID and the category value that is not yet available. The uninitialized values will begin initializing when requested for the first time or for the first time after not being used for a long time. When this list is populated, the HTTP status should be 206 to indicate partial content.

Example Response

```
{
  "results": [
    {
      "key": {
        "metricId": "K13CJXHV",
        "categoryValue": "John Doe(481)"
      },
      "value": {
        "lastUpdateTime": "2019-10-01T10:13:12.244-06:00",
        "value": false,
        "type": "BOOLEAN"
      }
    },
    {
      "key": {
        "metricId": "K0FLAEB6"
      },
      "value": {
        "lastUpdateTime": "2019-10-01T10:17:56.496-06:00",
        "count": 206,
        "type": "COUNT"
      }
    }
  ],
  "initializing": [
    {
      "metricId": "K08ZIKVN",
      "categoryValue": "Abraham Lincoln(871)"
    },
    {
      "metricId": "K08ZIKVN",
      "categoryValue": "Elvis Presley(121)"
    },
    {
      "metricId": "K1RNV20Q"
    }
  ]
}
```

METRIC VALUES

COUNT

```
{
  "key": {
    "metricId": "K0FLAEB6"
  },
  "value": {
    "lastUpdateTime": "2019-10-01T10:17:56.496-06:00",
    "count": 206,
    "type": "COUNT"
  }
}
```

BOOLEAN

```
{
  "key": {
    "metricId": "K08ZIKVN",
    "categoryValue": "John Doe(831)"
  },
  "value": {
    "lastUpdateTime": "2019-10-01T10:13:12.244-06:00",
    "value": false,
    "type": "BOOLEAN"
  }
}
```

STRING

```
{
  "key": {
    "metricId": "K0FLAEB6"
  },
  "value": {
    "type": "STRING",
    "lastUpdateTime": "2019-10-01T10:13:12.244-06:00",
    "value": "string value"
  }
}
```

DURATION

```
{
  "key": {
    "metricId": "K08ZIKVN",
    "categoryValue": "John Doe(831)"
  },
  "value": {
    "type": "DURATION",
    "lastUpdateTime": "2018-09-24T13:17:35.661-06:00",
    "finishedDuration": 20521,
    "numCountingUp": 2,
    "numCountingDown": 1
  }
}
```

EMPTY

```
{
  "key": {
    "metricId": "K08ZIKVN",
    "categoryValue": "John Doe(831)"
  },
  "value": {
    "lastUpdateTime": "2019-10-01T10:13:12.244-06:00"
  }
}
```

The duration in milliseconds is calculated using the following formula:

$\text{finishedDuration} + (\text{numCountingUp} - \text{numCountingDown}) \times (\text{atTime} - \text{lastUpdateTime})$

where *atTime* is the time (in milliseconds) you want to calculate the duration to.

```
AVERAGE_DURATION

{
  "key": {
    "metricId": "K08ZIKVN",
    "categoryValue": "John Doe(831)"
  },
  "value": {
    "type": "AVERAGE_DURATION",
    "lastUpdateTime": "2018-09-24T13:17:35.661-06:00",
    "durationSum": 20521,
    "count": 5,
    "numCountingUp": 2,
    "numCountingDown": 1
  }
}
```

The average duration in milliseconds is calculated using the following formula, provided that count is not zero:

$$\frac{\text{durationSum} + (\text{numCountingUp} - \text{numCountingDown}) \times (\text{atTime} - \text{lastUpdateTime})}{\text{count}}$$

where *atTime* is the time (in milliseconds) you want to calculate the duration to.

4.3. Creating a Realtime Metric

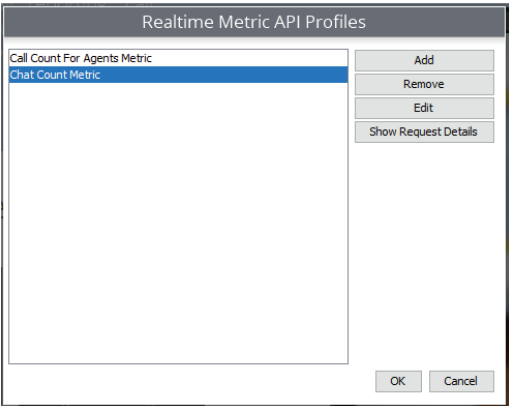
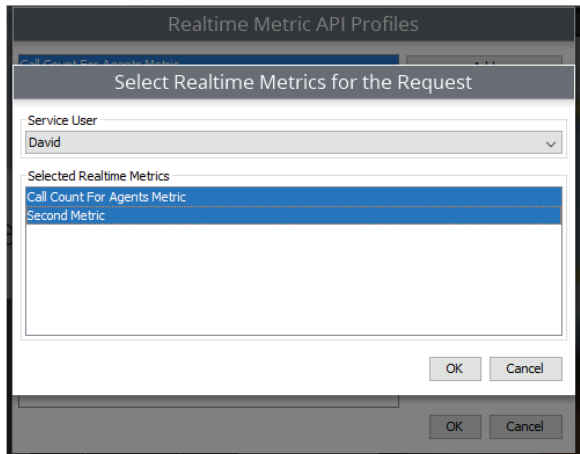
In order to poll a metric value, we'll need to first create it for API use. To do so, navigate to the Avaya Call Reporting main page. On the left sidebar, select API Enablement > Realtime Metric Access. If no Realtime metrics exist, then a pop up window will appear called "API Enabled Realtime Metric." Click "Show List View," which is located in the bottom left corner. This will allow us to see all of the reports that are currently enabled for API use. (If Realtime metrics do not exist, then you will instead open up to a pop up called "Realtime Metric API Profiles." Clicking "Add" in the top right corner will allow you to create a new metric value.)

A new pop up window called "API Enabled Report Profiles" will appear. Select the report you would like to see request details for. If the report that you want to run does not exist, click the "Add" button, which is located in the top right corner of the window. Fill out the information that the "Add" window asks, which includes:

- **Metric Name:** The title of the metric. This is what's displayed in the list view of all of the created Realtime metrics.
- **Metric Category:** Determines which category selections will be available.
- **Realtime Metric:** Select which Realtime value that you'd like to get results for.
- **Category Selection:** Choose which items to obtain values for based on the metric category selection. E.g., when we select "Agent" for the metric category, then the category selection will allow us to select which agents we want to obtain Realtime values for.
- **Service Account:** Select which service user has access to this metric.
- **Metric ID:** The ID associated with the metric. This is used in our API requests.
-
-

Once the information has been entered, click "OK." You should be redirected to the list view below. If not, then click API Enablement > Realtime Metric Access from the Avaya Call Reporting main menu and click "Show List View."

Once you've added metrics, click the "Show Request Details" button, which is located near the top right corner of the window. You'll then be prompted to select your service user and metrics.



Select the Realtime metric(s) you want and click "OK."

A new window will appear that provides you will all necessary information. It's also important to note that this window has two tabs. The "Metric Polling" tab will give you a list of API-enabled metric values. The "Metric Subscription" tab will give you Realtime data over a websocket connection.

POTENTIAL ERRORS

Status Code	Failure
400 Bad Request	Not all of the requested metrics are initialized. The metric values available will be returned in "results." The metric values that are not available will begin initializing, and their IDs will be in the "initializing" list. Try back in a few moments to get the values that are initializing.
401 Unauthorized	The authorization header wasn't valid.
403 Forbidden	The requested metrics are not assigned to the authenticated user.
404 Not Found	The requested metrics do not exist.

5.0. Recording Pause/Resume

After creating a Service User within the Avaya Call Reporting UI, you will be given an authentication key. Anytime a request is made to the API, an authentication key must be present.

By having the authorization key present, you will be able to access everything that the Service User is set up in Avaya Call Reporting to have access to.

```
ws://{AvayaCallReportingServerPath}/rest/api/v1/realtime-metric-subscription?id={MetricId}&auth_token={AuthenticationKey}
```

ARGUMENTS

AvayaCallReportingServerPath	The IP address and Port of the Avaya Call Reporting Server.
MetricId	The Metric ID of the desired metric to begin receiving updates for. A single websocket can subscribe to receive updates for multiple metrics. Pattern: ...?id={MetricId_1}&id={MetricId_2}...
AuthenticationKey	A Service User's authentication key.

RESPONSE

If a successful request is made, then the websocket will connect and will start receiving value update messages. There are several different types of value update messages that can be received. See examples of each type under the "Metric Values" section.

Example Response Message

```
{
  "key": {
    "metricId": "K0FLAE86",
    "categoryValue": "John Doe(481)"
  },
  "value": {
    "lastUpdateTime": "2019-10-01T10:17:56.496-06:00",
    "count": 15,
    "type": "COUNT"
  }
}
```

POTENTIAL ERRORS

Status Code	
400 Bad Request	Malformed or missing query parameters.
401 Unauthorized	The authentication key wasn't valid.
403 Forbidden	One or more of the requested metrics doesn't exist or are not assigned to the authenticated user.

6.0. Screen Pop Profile

A Screen Pop Profile will show external data to an agent on a pre-configured event. The screen pop can be configured to show events like “Call Presented,” “Agent Talking,” and “Chat Started.” When the configured event occurs, an external web page is presented to the agent.

USE CASES

Pop a 3rd party web page from the Contact Center Agent Client or the Java Desktop Client.

EXPECTED USER EXPERIENCE / WORKFLOW

- You research the target CRM and find out what URL and data is needed for the CRM page to pop correctly. This generally involves finding the base URL and query parameters that the CRM is expecting.
- Create a new “Screen Pop Profile” with the required parameters. Use the “Selected Role” setting to dictate which agent screens will pop.
- Use the Test option to ensure the CRM page opens correctly and properly reflects the parameters you configured.

SCREEN POP PROFILE CONFIGURATION

- **Profile name:** A name to identify the Screen Pop Profile
- **Agents whose screens will pop:** A role selection. When conditions are met, then those agents in the role will see their screen pop.
Pop the Screen when: This is the condition of when to pop an agent’s screen
- **Target URL:** This is the base URL of the pop. The user will see this URL on their screen when the condition is met.
- **Target Window:**
 - **Embedded (Agent Client):** This will pop within the CCAC (Contact Center Agent Client) and act like a new active media.
 - **External (New Window):** This will pop open a new tab to show the data. It will not act like a new active media in CCAC.
- **URL Parameters:** These are optional parameters that can be included with the request. These parameters can include call data from the call that meet the condition.
-

DEBUGGING

There is an Avaya Call Reporting Log Filter named “API Screen Pop” that can be used to find issues with the screen pop.

7.0. WebService Integration

This integration allows Avaya Call Reporting to send data to an external web service. On specific predefined events, Avaya Call Reporting will make an HTTP request to the external web service with information about the call or chat related to the event.

USE CASES

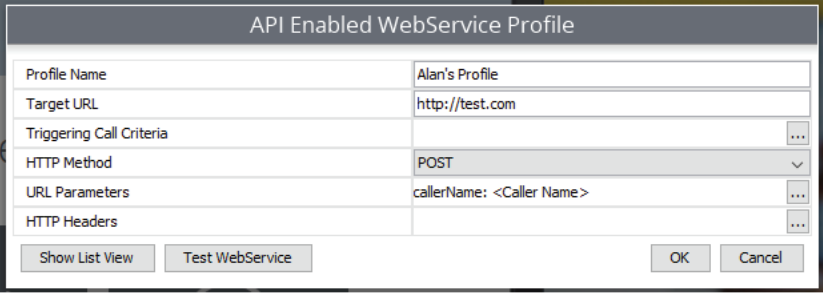
- Push a recording link to Salesforce each time a phone call is recorded
- Notify another system (like a CRM) to create a record each time a phone call finishes

EXPECTED USER EXPERIENCE / WORKFLOW

- You research the target CRM or Web Service and find out what URL and data is required for the POST or GET request. Create a new “Web Service Integration Profile” with the required parameters.
- Use the Test option to ensure the HTTP Request is generated correctly and handled properly by the target Web Service.
- Note: Requests originate from the Avaya Call Reporting server so the server must have the appropriate network and internet access.

○

7.1. Web Service Integration Profile Configuration



- **Profile Name:** A name to identify the API Enabled WebService Profile
- **Target URL:** This is the base URL of the request. Avaya Call Reporting will make a request to this URL if a call meets the given criteria
- **Triggering Call Criteria:** This will trigger when a call finishes. This list of criteria will be applied to the call to see if the call should trigger a request. If there are no criteria, then every call will trigger a request. If there are criteria and the call meets all the criteria, then the request will be triggered.
- **HTTP Method:** The HTTP method of the request
- **URL Parameters:** These are optional parameters that can be included with the request. These parameters can include call data from the call that meets the condition.
- **HTTP Headers:** Optional custom headers to be sent with the request

○

DEBUGGING

There is an Avaya Call Reporting Log Filter named “API Server Request” that can be enabled to find issues with the screen pop.

8.0. Web Chat Javascript API

These are javascript functions made available by Web Chat for custom integrations. All public API functions will be available on the `window._ximachat.api` object.

USE CASES

- Create your own Web Chat invitation
- Pass custom parameters (like Shopping Cart ID) from a customer's website interaction to the agent handling their chat.
 - Set the target routing skill based on customer context
-

8.1. Current Queue Status Request

```
_ximachat.api.getQueueStatus(onQueueStatus)
```

The `QueueStatus` will inform if there are agents ready for chats. This will fetch the current queue status of Multimedia chat queues from the server. It will execute the `onQueueStatus` callback function with the `QueueStates` object.

ARGUMENTS

<code>onQueueStatus</code>	A callback function that accepts the current <code>QueueState</code> .
Type: Function required	

Example QueueStates

```
{
  "Sales": { // Skill name
    "estimatedWaitTime": 100,
    "theQueueIsEmpty": true,
    "thereAreAgentsLoggedIn": false,
    "thereAreAgentsReady": false
  },
  "Support": { // Skill name
    "estimatedWaitTime": 100,
    "theQueueIsEmpty": true,
    "thereAreAgentsLoggedIn": true,
    "thereAreAgentsReady": true
  }
}
```


8.2. Subscribe to Queue Status

```
_ximachat.api.subscribeToQueueStatus(onQueueStatus)
```

The QueueStatus will inform if there are agents ready for chats. This will subscribe a callback function to queue state. Each time the QueueStatus changes on the server the onQueueStatus callback function will be executed.

ARGUMENTS

onQueueStatus	A callback function that accepts the current QueueState.
Type: Function required	

Example Queue States

```
{
  "Sales": { // Skill name
    "estimatedWaitTime": 100,
    "theQueueIsEmpty": true,
    "thereAreAgentsLoggedIn": false,
    "thereAreAgentsReady": false
  },
  "Support": { // Skill name
    "estimatedWaitTime": 100,
    "theQueueIsEmpty": true,
    "thereAreAgentsLoggedIn": true,
    "thereAreAgentsReady": true
  }
}
```

8.3. Set Custom Chat Parameters

```
_ximachat.api.setCustomParameters(customParameters)
```

Set custom chat parameters. These will be sent to display for the agent that is chatting with the external customer. They are intended to be used to send useful information to the agent chatting with the customer regarding the current page that the customer is on.

ARGUMENTS

customParameters	A list of custom chat parameters
Type: List of CustomParameter required	

Example CustomParameter

```
{
  "Name": "Page Title",
  "Value": "My Page That Has Chat"
}
```

8.4. Set Active Skill

```
_ximachat.api.setSkill(skill)
```

Set active Multimedia skill for chat. This can be used in place of the “URL Mapping” in the chat settings. Normally the URL mapping will decide what skill will be used for the chat based on the URL. If that is not possible, then this function can be used on the page to set the skill used for the chat.

ARGUMENTS

skill	The Multimedia skill for the chat.
Type: String required	

8.5. Get Active Skill

```
_ximachat.api.getSkill()
```

Get active Multimedia skill for chat. If the skill has not been set, then this will be undefined.

ARGUMENTS

Type: String	The Multimedia skill for the char or undefined.
--------------	---

8.6. Manually Start Chat

```
_ximachat.api.manuallyStartChat(skill)
```

Manually start a chat will an agent. This will open the form for the user to enter their information. Once the form is completed, the user will be placed in the queue and an agent will respond when available.

ARGUMENTS

skill	The Multimedia skill for the chat. If this was previously not needed or if left
Type: String optional	undefined, then the skill will be determined by the URL mappings.

8.7. Manually Start Xima Invite

```
_ximachat.api.manuallyStartChat(skill)
```

This is used if the “Show widget on page load” setting is set to “False.” This is intended to allow the developer of the page to decide when the chat can show and start, as well as to allow the developer to manually setting the skill.

ARGUMENTS

skill	The Multimedia skill for the chat. If this was previously not needed or if left
Type: String optional	undefined, then the skill will be determined by the URL mappings.

9.0. Web Chat Cloud Service Integration

Instructions on how to create a custom WebChat widget that uses Xima’s cloud backend.

To use this, you must request an API key. Please contact Xima to get an API key.

The URLs used in the Web Chat API begin with <https://mm-v1.ximasoftware.com>. Any breaking changes to the Xima Web Chat API will be deployed to <https://mm-v2.ximasoftware.com>, then mm-v3, etc.

ARGUMENTS

Create your own WebChat client that connects to Avaya Call Reporting Contact Center.

9.1. Checking a Session

```
GET /chat/queue?existingChatSessionId={chat-session-id}
```

An endpoint to verify that a chat session exists. The chat widget does this on page load.

ARGUMENTS

chat-session-id (optional)	A UUID to identify the chat session.
----------------------------	--------------------------------------

RESPONSE

If a chat session exists, a ChatSession object will be returned. If the chat session does not exist, a 204 response code will be returned.

RESPONSE BREAKDOWN

- chatQueueType: Possible values are SERVERTOCLIENT, CLIENTTOSERVER, or QUEUESTATUS
- id: Session ID
- installationId: The chat installation ID
- email: End user email
- name: End user name
- agent: Agent name
- agentImage: Base 64 encoded image

POTENTIAL ERRORS

Status Code	
204 No Conten	The chat session does not exist.

9.2. Starting a Session

POST /chat/queue

An endpoint to start a chat session and enter the chat queue. This requires a token received by completing a captcha.

ARGUMENTS

Send a ClientChatSessionRequest in the request body.

POST BODY BREAKDOWN

- installationId: The chat installation ID
- email: End user email
- name: End user name
- skill: The skill to queue for
- recaptchaToken: The token received after completing a captcha challenge
- customChatInfo: Some custom parameters that will show on the agent's screen

Example POST Body

```
{
  "installationId": "9c108bb3-8924-4234-a53c-be9ab284d4e3",
  "email": "endUserEmail@test.com",
  "name": "End User",
  "skill": "SkillName",
  "recaptchaToken": "9c108bb3-8924",
  "customChatInfo": [
    { "name": "Parameter Display Name", "value": "Parameter Value" }
  ]
}
```

RESPONSE

A ChatSession object will be returned.

RESPONSE BREAKDOWN

- chatQueueType: Possible values are SERVERTOCLIENT, CLIENTTOSERVER, or QUEUESTATUS
- id: Session ID
- installationId: The chat installation ID
- email: End user email
- name: End user name
- agent: Agent name
- agentImage: Base 64 encoded image

POST /chat/apiqueue

An endpoint to start a chat session and enter the chat queue. This uses an API key rather than a captcha token, therefore bypassing the captcha requirement.

ARGUMENTS

Send a ClientChatSessionRequest in the request body.

POST REQUEST BODY BREAK-

- installationId: The chat installation ID
- email: End user email
- name: End user name
- skill: The skill to queue for
- apiKey: The API key received from the Xima Software
- customChatInfo: Some custom parameters that will show on the agent's screen

Example POST Request Body

```
{
  "installationId": "9c108bb3-8924-4234-a53c-be9ab284d4e3",
  "email": "endUserEmail@test.com",
  "name": "End User",
  "skill": "SkillName",
  "apiKey": "9c108bb3-8924-4234-a53c-be9ab284d4e3",
  "customChatInfo": [
    { "name": "Parameter Display Name", "value": "Parameter Value" }
  ]
}
```

RESPONSE

A ChatSession object will be returned.

RESPONSE BREAKDOWN

- chatQueueType: Possible values are SERVERTOCLIENT, CLIENTTOSERVER, or QUEUESTATUS
- id: Session ID
- installationId: The chat installation ID
- email: End user email
- name: End user name
- agent: Agent name
- agentImage: Base 64 encoded image

Example Response

```
{
  "sessionDetails": {
    "serverToClientQueue": {
      "url": "https://sqs.us-east-1.amazonaws.com/775407092955/mm-queue-054d867a-76da-44f3-b63a-1ca187398f6d",
      "arn": "arn:aws:sqs:us-east-1:775407092955:mm-queue-054d867a-76da-44f3-b63a-1ca187398f6d",
      "created": "2012-04-23T18:25:43.511Z",
      "lastUsed": " 2012-04-23T18:25:43.511Z",
      "chatQueueType": "SERVERTOCLIENT",
    },
    "clientToServerQueue": {
      "url": "https://sqs.us-east-1.amazonaws.com/775407092955/mm-queue-054d867a-76da-44f3-b63a-1ca187398f6d",
      "arn": "arn:aws:sqs:us-east-1:775407092955:mm-queue-054d867a-76da-44f3-b63a-1ca187398f6d",
      "created": "2012-04-23T18:25:43.511Z",
      "lastUsed": " 2012-04-23T18:25:43.511Z",
      "chatQueueType": "CLIENTTOSERVER",
    },
  },
  "id": "9c108bb3-8924-4234-a53c-be9ab284d4e3",
  "installationId": "9c108bb3-8924-4234-a53c-be9ab284d4e3",
  "email": "endUserEmail@test.com",
  "name": "End User",
  "agent": "Agent Name", //agent name
  "agentImage": "iVBORw0KGgoAAAANSUHEUg...ruPfsj78onZEAAAAASUVORKSCYII=",
  "skill": "SkillName",
  "created": "2012-04-23T18:25:43.511Z"
}
```

9.3. Receiving Messages

Once a session has started, messages are sent to the chat widget over the `serverToClientQueue` defined in the `ChatSession` object. The `serverToClientQueue` is an AWS SQS queue. You can find documentation on reading messages off the queue [***here***](https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/sqs-examples-send-receive-messages.html).

POSSIBLE MESSAGES

The receivable message types are as follows:

ESTIMATED_WAIT_TIME

The estimated time left in queue until chat starts.

MESSAGE BREAKDOWN

- type: The message type
- estimatedWait: English description of wait time

CHAT_STARTED

Sent when an agent is ready to chat.

MESSAGE BREAKDOWN

- type: The message type
- agent: Agent name
- agentBase64Image: Base 64 encoded image for the agent

CHAT_TRANSFERRED

Sent when the end user is transferred to a new agent.

MESSAGE BREAKDOWN

- type: The message type
- newAgent: The name of the new agent
- newAgentImage: Base 64 encoded image for the new agent

CHAT_TEXT

Chat message sent by the agent

MESSAGE BREAKDOWN

- type: The message type
- text: The text of the chat

AGENT_TYPING

Sent when the agent is typing a message or when the agent has stopped typing.

MESSAGE BREAKDOWN

- type: The message type
- isTyping: This is whether the agent is typing or not. This option can be "True" or "False."

TAKE_SCREENSHOT

Sent when the agent would like to take a screenshot of the end user's browser window.

MESSAGE BREAKDOWN

- type: The message type

CHAT_ENDED

Sent when the agent ends the chat

MESSAGE BREAKDOWN

○ type: The message type

FORCE_CLOSE

Sent when the agent is unexpectedly disconnected from the chat. This ends the chat.

MESSAGE BREAKDOWN

○ type: The message type

MULTIMEDIA_DOWN

Sent when the server goes offline. This is the customer's server, not the server hosted at <https://mm-v1.ximasoftware.com>. This ends the chat.

MESSAGE BREAKDOWN

○ type: The message type

9.4. Sending Messages

Messages sent from the chat widget are sent to endpoints on <https://mm-v1.ximasoftware.com> as POST requests with various messages in the request body, not over an SQS queue like the incoming messages. All endpoints respond with either a 204 No Content or a 200 OK.

POST /chat/still-here

Needs to be sent every 30 seconds to keep the chat alive.

ARGUMENTS

Send a JSON message in the request body.

REQUEST BODY BREAKDOWN

○ chatSessionId: The chat session ID

Request Body Example

```
{
  "chatSessionId": "9c108bb3-8924-4234-a53c-be9ab284d4e3"
}
```

POST /chat

Sent when the end user sends a message to the agent.

ARGUMENTS

Send a JSON message in the request body.

REQUEST BODY BREAKDOWN

- chatSessionId: The chat session ID
- message: The text of the chat

Request Body Example

```
{
  "chatSessionId": "9c108bb3-8924-4234-a53c-be9ab284d4e3",
  "message": "I have a question about model no. 24601."
}
```

POST /chat/focus

Sent when the end user clicks or interacts with the chat text box. It notifies the agent that their message was read.

REQUEST BODY BREAKDOWN

Send a JSON message in the request body.

REQUEST BODY BREAKDOWN

- chatSessionId: The chat session ID
- focused: Indicates that the chat text box has been focused. This is always set to "True."

Request Body Example

```
{
  "chatSessionId": "9c108bb3-8924-4234-a53c-be9ab284d4e3",
  "focused": "true"
}
```

POST /chat/typing

Sent when user starts or stops typing.

ARGUMENTS

Send a JSON message in the request body.

REQUEST BODY BREAKDOWN

- chatSessionId: The chat session ID
- typing: A boolean to indicate whether the end user is typing or not

Request Body Example

```
{
  "chatSessionId": "9c108bb3-8924-4234-a53c-be9ab284d4e3",
  "typing": "true"
}
```

RESPONSE

The server will respond with a 204 No Content.

POST /screenshot/notify

Sent when the customer accepts or denies the agent's screenshot request.

ARGUMENTS

Send a JSON message in the request body.

REQUEST BODY BREAKDOWN

- chatSessionId: The chat session ID
- status: Whether the screenshot request is accepted or not. Possible values: ACCEPTED or REJECTED.
- uuid: The screenshot ID. Only sent if the screenshot is accepted. To get this UUID, see POST /screenshot

Request Body Example

```
{
  "chatSessionId": "9c108bb3-8924-4234-a53c-be9ab284d4e3",
  "status": "ACCEPTED",
  "uuid": "9c108bb3-8924-4234-a53c-be9ab284d4e3"
}
```

POST /screenshot

Used to upload a screenshot. The screenshot must be encoded as a base64 string. A screenshot UUID is returned.

ARGUMENTS

The POST request body should contain a base64 string representation of the image, prefixed by data:image/png;base64;. Currently the only supported file type is PNG.

POST /chat/end

Used to end the chat session. This request has no body. This also uses the browser session to find the relevant chatSessionId and ends the session.

10.0. Glossary

AUTHENTICATION

After creating a Service User within the Avaya Call Reporting UI, you will be given an authentication key. Anytime a request is made to the API, an authentication key must be present.

Add the following header to your API request:

```
Authorization : Bearer {authentication_key}
```

By having the authorization key present, you will be able to access everything that the Service User is set up in Avaya Call Reporting to have access to.

RECORDING STATUS

Possible recording statuses:

- PENDING - An event met a recording rule and should have a recording. Usually this means that the recording hasn't finished processing yet.
- SAVED - The recording is saved to an event on a call.
- DELETED_USER - The recording was manually deleted by a user.
- DELETED_SPACE - The recording was deleted by the Recording Library because the Recording Library was low on space.
- DELETED_TIME - The recording was deleted by the Recording Library based on the Recording Library's time retention policy.
- DELETED_UNKNOWN - The recording was saved, but was manually deleted from the filesystem.
- PENDING_DELETE_SPACE - The recording is flagged to be deleted later because of the Recording Library's low space.
- PENDING_DELETE_TIME - The recording is flagged to be deleted later because of the Recording Library's retention policy.
-
-

RECORDING KEY

The database key for a recording.

Example: 89e08e9f-a2aa-4059-9f93-1f53e3719431-1568754442049

Format: {recording-UUID}-{startTime}.

- {recording-UUID}: The recording's UUID, e.g., "89e08e9f-a2aa-4059-9f93-1f53e3719431"
- {startTime}: Start of the recording in epoch millis.

CALL KEY

The database key for a recording.

Example: 89e08e9f-a2aa-4059-9f93-1f53e3719431-1568754442049

Format: {recording-UUID}-{startTime}.

- {recording-UUID}: The recording's UUID, e.g., "89e08e9f-a2aa-4059-9f93-1f53e3719431"
- {startTime}: Start of the recording in epoch millis.

AGENTID

Agent Identifier. Valid identifiers:

- The agent extension. Example: "203"
- The agent name and extension. Example: "Nate Thatcher(203)"
- User defined alias.

There will be a setting in Avaya Call Reporting to define the mapping from alias to user. This alias is a custom string entered by the user.