



# **Avaya Open Interfaces**

Communications Control Toolkit REST SDK

Overview of Avaya Open Interfaces CCT REST API.....	3
Rationale for Supporting REST API.....	3
Revision Information .....	4
Changes in 7.0:.....	4
Advantages of SOA.....	5
SDK contents .....	5
Tutorial.....	6
Reference Client.....	6
SDK support.....	6
Related Documents .....	7
Contacting support .....	8
Programming with the CCT REST Open Interface API.....	9
CCT User Session Resource .....	9
Agent Resource .....	9
Contact Resource .....	9
Subscriptions Resource .....	9
Reference Information .....	10
FAQs .....	10
Call Attached Data Limit .....	10
Intrinsics Limitations .....	10
Security .....	10
User Validation & Authentication.....	10
Encryption:.....	11

## Overview of Avaya Open Interfaces CCT REST API

The Avaya Open Interfaces CCT REST<sup>1</sup> API is a set of REST-based web services enabling 3<sup>rd</sup> party applications to integrate with the Avaya Aura™ Contact Center. The REST web services complement the set of SOAP<sup>2</sup>-based Avaya Open Interfaces CCT web services, currently providing a subset of their functionality. They are hosted on the Avaya CCT server and require that the CCT SOA<sup>3</sup> settings be configured.

REST-based web services provide access to functionality directly through the exchange of HTTP messages. Resources such as agents and contacts are directly identified by a URI. To retrieve the state of a resource a HTTP GET is sent. To modify the state of a resource, a HTTP POST message is sent along with data representing the desired state.

The web services can be optionally configured to use TLS<sup>4</sup>. If this functionality is enabled the customer will be required to provide the necessary certificates. Refer to the Avaya Aura™ Contact Center Commissioning Guide in the section *Related Documents* for details on how to configure Open Interfaces on the CCT server.

The REST services are published when SOA is configured and enabled on a CCT server. Services are available at the following URL where <CCT HostName> is the host name of the CCT server: `http://<CCT HostName>:9085`

### ***Rationale for Supporting REST API***

The motivation for supporting REST web services in addition to traditional SOAP web services is to support the fast development of lightweight rich applications on the Avaya Aura Contact Center platform.

Calls to the REST API are typically lightweight and are easier to embed in web applications code in HTML/JavaScript than the SOAP API. The REST interface consists fundamentally of URIs with HTTP calls.

In their simplest form REST calls can be made directly from a standard web browser such as Microsoft Internet Explorer® or Mozilla Firefox®. Due to this lightweight nature, the REST API can be targeted at pure thin-client development and Web 2.0 developers.

---

<sup>1</sup> REST is a programming style for web services that constrain the interface to a set of standard operations like GET and POST. It is based on the REST architectural style of the World Wide Web as described by Roy Fielding in 2000.

<sup>2</sup> SOAP is an industry-accepted W3C specification used to describe messages (XML documents and their attachments) so that they can be sent across a network.

<sup>3</sup> A Service Oriented Architecture (SOA) is a collection of services. A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services. These services communicate with each other

<sup>4</sup> Transport Layer Security (TLS). Successor to SSL. TLS is an IETF standards track protocol, last updated in RFC 5246.

In contrast to REST, using the SOAP API requires more complexity to handle the request and generate the response but offers a greater range of extensions and stronger tool support. Additionally SOAP provides better support for sending attachments such as binary data.

While encryption of messages in REST depends on the use of SSL<sup>5</sup> or TLS at the transport layer, WS-Security<sup>6</sup> for the SOAP API allows for more fine-grained message-level encryption.

Unlike the SOAP-based web services, the REST web service interfaces are not described using WSDL<sup>7</sup>. Instead the interface is documented as a standalone wiki-style web page, *Avaya\_REST\_API.html* that accompanies this SDK.

## ***Revision Information***

### **Changes in 7.0:**

- SDK now targets Java 1.8 instead of 1.6
- Updated version numbers in documentation
- Added revision information

---

<sup>5</sup> Secure Socket Layer (SSL). Transport layer security protocol for communication over a network such as the internet, originally developed by Netscape.

<sup>6</sup> Web Services Security (WS-Security) is an extension to SOAP to apply security to Web services. It is a member of the WS-\* family of web service specifications, published by OASIS.

<sup>7</sup> Web Services Description Language. WSDL provides a grammar for describing services as a set of endpoints that exchange messages. A WSDL document serves as a language and platform-agnostic (XML) description of one or more services. It describes the services, how to access them, and what type of response (if any) to expect.

<sup>8</sup> The Web Application Development Language (WADL) is undergoing development as a RESTful equivalent of WSDL. To date this is not an industry standard.

### What are the Open Interfaces?

The Open Interfaces are installed on your Avaya Communication Control Toolkit server but require an associated Open Interfaces license before they can be enabled, see *Avaya\_Contact\_Center\_Commissioning* for configuring services.

These services supplement your existing tools to simplify communication control development. This documentation is for the CCT REST Open Interfaces which are related to the following version of the contact center.

Version
Avaya Aura Contact Center 7.0

Please refer to the Developer Program web site for the latest version of all documentation.

## ***Advantages of SOA***

SOA is an architectural style whose goal is to achieve loose coupling among interacting software agents. This means that applications existing on different Operating Systems written in different programming languages can communicate and exchange data with each other.

The Open Interfaces provides a number of basic features that make it easier to deploy a variety of applications. These features include:

- **Zero deployment** – SOA has a loose coupling of services with operating systems, programming languages and other technologies and hence does not place additional requirements on the client to download Avaya libraries
- **Technology agnostic** – SOA is not associated with a specific OS, programming language or technology and hence leaves the clients free to develop in their language(s) of choice.
- **Common integration point** – SOA provides a common integration technology between different technologies and applications.

Comparison of REST to SOAP API

## ***SDK contents***

The SDK contains the following elements:

- Application programming interface documentation
- Tutorial for creating a client using simple HTML and Javascript technology
- Full reference client – self-contained HTML/Javascript Web page.

## Tutorial

A tutorial is provided in the folder: [tutorial](#). It contains detailed steps to create a Web-page based CCT REST OI client that can carry out the following operations:

- Log a CCT user in to Contact Center
- Subscribe for event notifications on a terminal
- Make a call
- Drop a call

Based on the tutorial steps, users can extend the client for Answer Call, Single-step Transfer and Supervised Transfer operations. A full client implementation including these operations is provided by the reference client.

## Reference Client

The CCT REST Open Interfaces contains a reference client that can be used to test the RESTful web services. The client uses only HTML and Javascript and demonstrates how a simple standalone web page can be constructed to access Open Interface operations of the Avaya Aura™ Contact Center.

The source for the reference client is available at:

[\ReferenceClient\CCT\\_REST\\_OI\\_RefClient\\_v0.07.html](#)

This reference client requires a Web server such as Apache Tomcat or Jetty to be running. A sample Web server is included with the tutorial package. Once a Web server is running, the reference client can be launched by hosting the Web page on the server and opening the page in a browser (also included in the tutorial bundle).

**NOTE:** The browser must support *cross-domain requests*<sup>9</sup>. Microsoft Internet Explorer 8® (and later) and Mozilla Firefox 3.5® (and later) are examples of such browsers.

## SDK support

Support for the SDK APIs is supplied by the Developer Partner Program. For more information about contacting support personnel, see *Contacting Support*.

---

<sup>9</sup> Cross-domain HTTP requests are HTTP requests for resources from a different domain than the domain of the resource making the request. See <http://msdn.microsoft.com/en-us/library/dd573303%28VS.85%29.aspx> and [https://developer.mozilla.org/en/HTTP\\_access\\_control](https://developer.mozilla.org/en/HTTP_access_control) for further information.

***Related Documents***

The following table lists documents that contain more information about the Open Networking Control Toolkit:

<b>For information about</b>	<b>See</b>
Planning and engineering requirements	Avaya Aura™ Contact Center Planning and Engineering (NN44400-210)
Installing, upgrading, and maintaining the Communication Control Toolkit (CCT) server software	Avaya Aura™ Contact Center Installation (NN44400-311)  Avaya Aura™ Contact Center Commissioning (NN44400-312)  Avaya Aura™ Contact Center Upgrade and Patches (NN44400-410)
Troubleshooting problems	Avaya Aura™ Contact Center Troubleshooting (NN44400-712)

## Contacting support

If you are a DevConnect member, contact the Avaya Dev Connect personnel through the following e-mail address: [devconnect@avaya.com](mailto:devconnect@avaya.com).

For details about becoming a DevConnect Member, see:

<http://www.avaya.com/gcm/master-usa/en-us/corporate/alliances/devconnect/index.htm>.



## Programming with the CCT REST Open Interface API

The Avaya CCT OI REST API is based on the use of resources that are uniquely identified by URI. Examples of resources are: agent, CCT user, and contact.

Full details of the API are available at the self-contained wiki page at:

[\Avaya CCT REST API.html](#)

The REST API operations provided by the Avaya CCT OI REST API are organized by the resource to which they apply. The operations below are currently supported and detailed on the wiki page

### ***CCT User Session Resource***

Log in a CCT user

Log out a CCT user

### ***Agent Resource***

Log in an agent to a terminal

Log out an agent to a terminal

### ***Contact Resource***

Make a call

Answer a call

Release a call

Do a single-step call transfer

Initiate a supervised call transfer

Complete a supervised call transfer

Add attached data to a call

Add intrinsics to a call

### ***Subscriptions Resource***

Create a subscription for event notifications

## Reference Information

This information is common with the Open Interfaces (SOAP) CCT SDK.

### ***FAQs***

For a complete and up to date list of Frequently Asked Questions please visit and register on the Avaya DevConnect site.

### ***Call Attached Data Limit***

There is maximum limit of 45k of data to be attached to a contact, please refer to the CCT documentation for further information.

### ***Intrinsics Limitations***

It is not recommended that intrinsics should be used to store large quantities of data as it may impact overall performance in the Contact Center. If these intrinsics are to be accessed using the scripting engine contained in Contact Center then there are additional limitations to the size and amount of intrinsics that can be associated with a contact. please refer to the scripting documentation for further details.

### ***Security***

There are two levels of security for the CCT Open Interfaces. Firstly the web services authenticate whether the user is know and configured and secondly all communication between client and server can be optionally encrypted.

### ***User Validation & Authentication***

Before the service can be used the developer (CCT User) must first receive a Single-Sign-On (SSO) token that will be used for subsequent calls to the REST API. This is carried out by doing a CCT User login.

When the user submits their username and password to the service it is validated to determine if it is a valid windows user. Depending on the initial configuration of the service this user can be a local windows user on the CCT Server or a windows user on the domain.

Once the service has verified that the user is a valid windows user it then checks to see if the user is configured in the CCT console. Once both of these criteria are met the user receives an SSO token representing their current session.

Each session has a time out associated with it. The default for a Session timeout is 2 hours but can be configured using the OI configuration screen. Each time the service is accessed this timeout is reset.

For further information on how to configure the user store and CCT users please consult the section Avaya Contact Center Commissioning.

**Encryption:**

The client communicates with the server through the use of HTTP messages. The Open Interfaces CCT can be configured to encrypt these messages at the transport layer using TLS configuration.

To configure TLS an administrator will need to generate a certificate signing request (CSR). The administrator will need to present this CSR to their chosen Certificate Authority (CA) to receive a corresponding certificate.

For further information on how to configure TLS please consult the document Avaya Contact Center Commissioning.