Finesse APIs: Getting started with the REST APIs and XMPP events

Denise Kwan, Software Engineer @ DevNet
Cisco Spark

Questions?
Use Cisco Spark to communicate with the speaker after the session

How
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space

cs.co/ciscorobot#DEVNET-1024
Agenda

- Introduction to Finesse
- Finesse REST APIs and Notifications
- Understanding Finesse’s Developer’s Guide
- Hands-on Workshop
Cisco Finesse
Revolutionizing the Contact Center Agent Experience

- Browser-based agent desktop for easy management & upgrades
- Empower agents via a personalized all-in-one application
  - Integrate Finesse into existing applications (thick or thin)
  - Integrate existing applications into the Finesse agent desktop
- Easy-to-use REST and JavaScript APIs for performing agent and supervisor actions programmatically
New in Finesse 11.6

• Integrated System generated Not Ready Reason Codes
• Secondary CallId for transfers and conferences
• Alternate host for loading gadgets
• State and call history gadgets
• Direct Transfer
• Ability to make a call from READY state
• Ability to monitor Outbound calls on ACD line

Reserve the UCCX 11.6 sandboxes to try out these new features!
Finesse REST APIs and Notifications
REST APIs

• Modeled after HTTP

• Works in thick and thin client integrations

• Easy to integrate into applications – there are many libraries to make HTTP requests

• Used to request an action

• GET requests are synchronous. The response body of a successful GET request contains all requested contents.

• PUT and POST requests are asynchronous. A successful response is an HTTP return code of 200 or 202. The response body does not contain the updated object information.
  • When the PUT and POST request is processed, a notification via XMPP will be sent with the updated object information
Finesse Notifications

• Uses XMPP protocol as the data communication channel

• For web applications, the XMPP payload needs to be encapsulated within the HTTP protocol, known as BOSH

• Can find existing XMPP libraries here: https://xmpp.org/software/libraries.html

• Used for asynchronous reception of events

• There can only be 1 connection to the Finesse Notification Service per user, where the Finesse out of the box desktop counts as one.
2) Application/browser calls the User Sign In REST API

```
PUT http://finesse.cisco.com:8082/finesse/api/User/Agent001
Content-Type: application/xml
Request Body:
<User>
  <state>LOGIN</state>
  <extension>6001</extension>
</User>
```

3) Finesse server receives the request and immediately sends back a HTTP response of 202 Accepted with a blank response body

```
HTTP/1.1 202 Accepted
```

4) Finesse publishes the response of the request as a notification to the User node /finesse/api/User/Agent001

```
<Update>
  <data>
    <apiErrors>
      <apiError>
        <errorData>260</errorData>
        <errorMessage>CF_INVALID_DEVICE</errorMessage>
        <errorType>Invalid Device</errorType>
      </apiError>
    </apiErrors>
  </data>
  <event>PUT</event>
  <source>/finesse/api/User/Agent001</source>
</Update>
```

5) The client receives the notification via the BOSH connection and processes the response notification

```
HTTP/1.1 202 Accepted
```

1) Agent clicks the Sign In button
Understanding Finesse’s Developer’s Guide
On DevNet’s Finesse site:

- On your browser, go to the DevNet home page: http://developer.cisco.com
- Click on the Technologies tab
- Click on Collaboration on the left hand side and select Finesse under Contact Center
- Click on Docs on the upper right
- Select REST API Developer Guide under Guides

What is Finesse?

Cisco Finesse is a next-generation agent and supervisor desktop designed to provide a collaborative experience for the various communities that interact with your customer service organization. It
How to use the Finesse Developer Guide

The URI is the REST API URL where <FQDN> is the fully-qualified domain name of the Finesse server.

In a UCCE deployment, you must include the port number 8445 in the URI for HTTPS requests. In a UCCX deployment, you must include the port number for both HTTP (8082) and HTTPS (8445).

The <id> is the user’s AgentID (Peripheral number) for UCCE and User ID for UCCX.

The HTTP Method is the HTTP request method/verb for this REST API.

- GET: Retrieve a single object or list of objects (for example, a single user or list of users).
- PUT: Replace a value in an object (for example, to change the state of a user from NOT_READY to READY).
- POST: Create a new entry in a collection (for example, to create a new reason code or wrap-up reason).
- DELETE: Remove an entry from a collection (for example, to delete a reason code or wrap-up reason).

The Security Constraints states which user/role has permissions to use this REST API.

In this example, the user <id> is the only user who can use this REST API.

The Content Type is the MIME type of the body of the request (used with POST and PUT requests).

All Finesse REST APIs use Application/XML as the content type.

The Input/Output Format is the format of the HTTP request/response body.

All Finesse REST APIs use XML as the input/output format.

The HTTP Request is the content of the HTTP request body (used with POST and PUT requests). Since the input format of the REST API is XML, the content of the body is in XML format.
The Request Parameters are the parameters for this REST API. There are three different types of parameters:

- Body parameter: A parameter (also known as a complex parameter) that appears in the body of the message
- Path parameter: A parameter that is included in the path of the URI
- Query parameter: A parameter is passed in a query string on the end of the URI you are calling

It also defines each parameter and states if it is required or optional.

The Example Failure Response is a sample HTTP response body when the REST API call was not successful.

With the exception of the SystemInfo API, all Finesse REST APIs use HTTP BASIC Authentication which requires the credentials to be sent in the "Authorization" header. The credentials contain the username and password, separated by a single colon (:), within a BASE64-encoded string.

The HTTP Response lists the different HTTP status codes and its definition that can be returned for this REST API.

This is useful for handling the different types of failures associated to this REST API.

The Notifications Triggered lists the object(s)/node(s) that will receive a notification as a result of this REST API.
Hands-on Workshop
Objective

- Use Postman to make REST API requests
- Use Adium to receive Finesse notifications
- Understand how the REST APIs and Finesse notifications work hand in hand
  - Make a REST API request and receive the notification(s) as a result of the request
Set up Adium to receive Finesse notifications

- **Adium** is an instant messaging application for Mac OS X that can connect to XMPP (Jabber).
- **Pidgin** is equivalent and the preferred client for Windows.

There are other XMPP clients, but not all of the clients expose an XMPP console, which we will need to view the Finesse notifications.

Adium is just a tool to view the Finesse notifications. This is not needed if you are programmatically receiving notifications.

Go to [https://xmpp.org/software/libraries.html](https://xmpp.org/software/libraries.html) for a list of XMPP libraries for the different programming languages.
Automatically subscribed nodes

- Each agent is automatically subscribed to the following notification feeds, where {id} represents the agent ID for that agent:
  - User - /finesse/api/User/{id}
  - Dialogs - /finesse/api/User/{id}/Dialogs
  - Queues - /finesse/api/Queue/{id}
  - Media - /finesse/api/User/{id}/Media/{mrd-id}
  - SystemInfo - /finesse/api/SystemInfo

- Any notifications sent to these nodes will automatically be received by that agent if they connected to the Finesse Notification Service.
User

• The User object represents an agent or supervisor and includes information about the user, such as roles, state, and teams.

• In this step, we will be using two User APIs:
  • Get the User object:
    GET http://hq-uccx.abc.inc:8082/finesse/api/User/Agent001
  • Log in the User:
    PUT http://hq-uccx.abc.inc:8082/finesse/api/User/Agent001
    <User>
      <state>LOGIN</state>
      <extension>6001</extension>
    </User>
Dialog

• Make a call:

  **POST** http://hq-uccx.abc.inc:8082/finesse/api/User/Agent001/Dialogs

  <Dialog>

  <requestedAction>MAKE_CALL</requestedAction>
  <fromAddress>6001</fromAddress>
  <toAddress>6002</toAddress>

  </Dialog>
Dialog

- Take an action on the call:

  PUT http://hq-uccx.abc.inc:8082/finesse/api/Dialog/{dialog id}

  <Dialog>
    <targetMediaAddress>6001</targetMediaAddress>
    <requestedAction>DROP</requestedAction>
  </Dialog>
User

- Log out the User:

  ```xml
  PUT http://hq-uccx.abc.inc:8082/finesse/api/User/Agent001
  <User>
    <state>LOGOUT</state>
  </User>
  ```
How to Learn More About Finesse

• At Cisco Live!
  • Finesse Learning Labs

• Finesse DevNet site: http://developer.cisco.com/site/finesse
  • Finesse Community (forum)
  • Finesse Sample Gadgets
  • REST API & JavaScript library Developer Guides
  • Learning Labs
  • Finesse Sandbox (PCCE & UCCX)
Cisco Spark

Questions?
Use Cisco Spark to communicate with the speaker after the session

How
1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Spark or go directly to the space
4. Enter messages/questions in the space

cs.co/ciscolivebot#DEVNET-1024
• Please complete your Online Session Evaluations after each session

• Complete 4 Session Evaluations & the Overall Conference Evaluation (available from Thursday) to receive your Cisco Live T-shirt

• All surveys can be completed via the Cisco Live Mobile App or the Communication Stations

Don’t forget: Cisco Live sessions will be available for viewing on-demand after the event at CiscoLive.com/Online.
Continue Your Education

• Demos in the Cisco campus
• Walk-in Self-Paced Labs
• Tech Circle
• Meet the Engineer 1:1 meetings
• Related sessions
Thank you