

Example API and Toolkit README

- [Introduction](#)
- [Requirements](#)
- [Getting Started](#)
 - [Building the Toolkit](#)
 - [Examples](#)
 - [Agent Skill Group Demo \(Packaged CCE Only\)](#)
 - [Agent Attribute Demo \(Packaged CCE and Unified CCE\)](#)
 - [Initialize Demo \(Packaged CCE Only\)](#)
 - [Remote Site Demo \(Packaged CCE Only\)](#)

Introduction

The purpose of the unifiedconfig example toolkit is to provide an easy means of programmatically interacting with the unifiedconfig REST web service in Java. The examples provided include:

- A simple client that handles authentication, making requests, and receiving responses.
- Classes to handle interacting with the Agent, Skill Group, and Attribute APIs.
- Examples of creating, associating, and deleting Agents, Skill Groups, and Attributes.
- Sample code that automates the steps performed in Configure Deployment workflow to initialize fresh installs.

Requirements

- Java 1.7.0_45 or future 1.7 update
- Apache Maven 3.2.1
- Access to a 11.0(1) or greater Unified CCE Data Server or External HDS

Getting Started

The unifiedconfig example toolkit is downloadable as a zip file from Cisco's DevNet website. The zip file contains a maven project and example source code for interacting with the REST service.

Building the Toolkit

To build the Toolkit, simply use the maven install command from the base level of the project:

```
"mvn clean install"
```

Note: After building the toolkit from the base directory, change directory to webconfig-api-toolkit-samples (cd webconfig-api-toolkit-samples) before launching the example programs from command line.

Examples

Agent Skill Group Demo (Packaged CCE Only)

The AgentSkillGroupDemo class is in the examples package.

To launch the program from the command line (passing in the real IP, username, and password for a Unified CCE Data Server), **type:**

```
mvn exec:java -Dexec.mainClass="com.cisco.ccbu.cce.unifiedconfig.toolkit.examples.AgentSkillGroupDemo" -Dexec.args="10.86.135.193 administrator@test.com abc123="
```

The demo performs the following operations:

1. Creates three Agents and one Skill Group.
2. Performs a GET operation on an Agent URL that does not exist and prints out the error.

3. Updates one Agent to include the Skill Group.
4. Performs two list functions: one with search parameters and one without search parameters.
5. Deletes the four objects.

Agent Attribute Demo (Packaged CCE and Unified CCE)

The AgentAttributeDemo class is in the examples package. (If running under Unified CCE, at least one agent must exist before launching the test.)

To launch the program from the command line (passing in the real IP, username, and password for a Unified CCE Data Server or External AW-HDS), **type**:

```
mvn exec:java -Dexec.mainClass="com.cisco.ccbu.cce.unifiedconfig.toolkit.examples.AgentAttributeDemo" -Dexec.args="10.86.135.193 administrator@test.com abc123="
```

The demo performs the following operations:

1. Creates one Agent (or looks up one Agent if Unified CCE).
2. Creates one Attribute.
3. Performs an update operation on the Agent to associate it with the Attribute and give it a value.
4. Perform an update operation on the Agent to remove the association and re-associate to any previously existing attributes.
5. Deletes created objects: Agent and Attribute for Packaged CCE, or just Attribute for Unified CCE.

Initialize Demo (Packaged CCE Only)

The Initialize demo is in the examples package. This demo program allows you to initialize a fresh installation without stepping through the Configure Deployment workflow in Unified CCE Administration.

Set up

1. Edit the file "initialize.properties".
2. Ensure all values are filled in properly.

To launch the program from the command line, type:

```
mvn exec:java -Dexec.mainClass="com.cisco.ccbu.cce.unifiedconfig.toolkit.examples.InitializeDemo"
```

The initialization procedure begins immediately.

Notes

- Ensure that "pcce.host" points to the Side A Unified CCE Data Server.
- Before running the demo program, you must have installed the Packaged CCE components as described in the "Installation" section of the *Cisco Packaged Contact Center Enterprise Installation and Upgrade Guide*, configured Microsoft SQL Server for the Side A and B Unified CCE Data Servers, and configured Domain Manager on one of the Data Servers.
- Do not run Web Setup for any Unified CCE components before running the demo program.
- Packaged CCE is not ready to use until you complete the post-initialization configuration procedures that are documented in the *Cisco Packaged Contact Center Enterprise Installation and Upgrade Guide*.
- The demo program does not support an off-box Unified Communications Manager, and needs to be adapted for that functionality.
- Error checking is minimal.

Remote Site Demo (Packaged CCE Only)

The remote site demo is in the examples package. This demo program allows you to create, update and delete a remote site without stepping through the Remote Site wizard workflow in Unified CCE Administration.

Set up

1. Edit the file "site.properties".
2. Ensure all values are filled in properly.

To launch the program from the command line, type:

```
mvn exec:java -Dexec.mainClass="com.cisco.ccbu.cce.unifiedconfig.toolkit.examples.RemoteSiteDemo"
```

The remote site creation begins immediately. The remote site will be created with Agent and Multichannel PG, which will be updated with VRU PG then. The site will be deleted at the end after asking for confirmation.

Notes

- Ensure that "pcce.host" points to the Side A Unified CCE AW.
- Before running the demo program, you must have installed and configured the Packaged CCE. Also you must have installed a remote site PG(having Agent, VRU and Multichannel) and the associated peripherals.
- Error checking is minimal.

Accessing Additional APIs

Build additional APIs using this same methodology, by following the *Cisco Packaged Contact Center Enterprise Developer Reference Guide*.

For each new API, follow these steps:

1. Create a class that extends BaseApiBean.
2. Add the annotations @Path("<path suffix>") @XmlElement(name = "<bean name>") where path suffix is the lowercase suffix to the unified config URL and bean name is the camel case root XML element for the API.
3. Add in the attributes with get and set methods for API. Use the ReferenceBean class for all references (see AgentDeskSettings and SkillGroups in Agent for examples).
4. Create a class that extends the BaseApiListBean and implement the methods getItems and setItems methods, and add the @Path and @XmlElement annotations on the class and the @XmlElementWrapper and @XmlElement annotations on the getItems method.