cucm-http-api documentation

Release 1.0

Mobile Oxford team, IT Services, University of Oxford

October 05, 2015

Developer

1.1 Overview

This API keeps as much as possible the same terminology than the one used in the WSDL from CUCM.

1.1.1 Dropwizard

This application uses Dropwizard, it is recommended to be familiar with their documentation.

1.1.2 Code architecture

See Javadoc for more information.

1.2 Configuration

Configuration is done through a YAML file containing various sections.

An example is available at `src/main/resources/configuration.yml.example`.

1.2.1 Section "cucm"

This section is about credentials to connect to a CUCM server.

- `host` is the host name
- `user` is the user to log in as
- `password` is the password for the `user`

1.2.2 Section "apiauth"

This section is about credentials to connect to this application using HTTP basic auth.

- `user` is the name of the user that will have to be use when attempting to do a basic auth
- `password` is a SHA-256 hash of the password

The hash of the password can be generated with `echo -n "password" | shasum -a 256`.

1.2.3 Section "logging" (optional)

Configure the logging of the application (by overriding the default configuration), see Dropwizard's documentation.

1.2.4 Section "http" (optional)

Configure the web server (by overriding the default configuration), see Dropwizard's documentation.

See Security considerations for more recommendations on this section.

1.3 Deploying the application

The application is built as a single JAR using `mvn install`, this JAR embeds a Jetty web-server.

The application can run as `java -jar cucm-http-api.jar server path/to/configuration.yml`.

The process can be runned using Supervisor or Circus to be properly managed.

1.4 Security considerations

This software has been designed to be used as a technical API (used as a middleware) exposed to a limited audience.

The HTTP API exposes a limited set of values from CUCM, some of them are sensitive, in particular the "phone name" (which is the unique identifier of a device on a network).

All methods should be protected by authentication.

1.4.1 Design

This software does not store any data and is completly stateless.

1.4.2 Recommendations

Although the exposed HTTP API is protected by HTTP Basic Auth, it is recommended to use an appropriate firewall configuration to limit the exposure of data.

If all applications using this software are on the same machine, it should be bind to a local interface (see configuration).

1.4.3 Protecting resources

See Dropwizard auth documentation for more details, resources should include a parameter annotated `@Auth` as shown in the example below.

```
@GET
public List<Phone> get(@QueryParam("dirn") IntParam dirn, @Auth User user) {
```

HTTP API

2.1 Phone endpoint

Endpoint for operations on phones.

All these calls require basic auth.

GET /phone/
Get phones corresponding to a directory number

Example request:

```
GET /phone/?dirn=123 HTTP/1.1
Host: 127.0.0.1
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

[
  {
    "product": "Cisco 7942",
    "description": "description",
    "model": "Cisco 7942",
    "name": "uniquename",
    "uuid": "{UUID}",
    "dirns": [
      "123"
    ],
    "speeddials": [
      {
        "index": "1",
        "dirn": "11",
        "label": "office"
      },
      {
        "index": "2",
        "dirn": "22",
        "label": "home"
      },

    ]
```

```
        }
    ]
```

Query Parameters

- **dirn** – directory number to search for

Status Codes

- 200 OK – list of phones matching the directory number (empty list if no phone)

2.2 Speeddials endpoint

Endpoint for operations on speeddials.

All these calls require basic auth.

POST /speeddials
Update the speeddials for a phone by its name.

Example request:

```
POST /speeddials?phone=name HTTP/1.1
Host: 127.0.0.1
Accept: application/json
Content-Type: application/json
[
    {
        "index": "1",
        "dirn": "11",
        "label": "office"
    },
    {
        "index": "2",
        "dirn": "22",
        "label": "home"
    },
]
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: plain/text
{uniqueid}
```

Query Parameters

- **phone** – unique name of the phone to update

Status Codes

- 200 OK – request done
- 400 Bad Request – Bad request (if you don't pass the parameter)
- 500 Internal Server Error – an exception occured

CUCM calls

3.1 Overview

List of methods used in this application and consequences on CUCM AXL Service.

Method	Action	Uses
Get phone info	Read	SQL, AXL
Update speed dials	Update	AXL

3.1.1 Get phone info

Used in Phone endpoint, it does two **read-only** operations:

1. SQL query to find names of phone matching a directory number

2. Get detailed information about every phone, by its phone name (GetPhoneReq)

3.1.2 Update speed dials

Used in Speeddials endpoint, it does one **update** operation:

1. Do an **UpdatePhoneReq** for a given phone name

API

4.1 Javadoc

4.1.1 uk.ac.ox.it.cha

CucmHttpApiService

class **CucmHttpApiService** extends Service<*AppConfiguration*>
Main entry point of the application

Author martinfilliau

Methods

initialize
public void **initialize** (Bootstrap<*AppConfiguration*> *bootstrap*)

main
public static void **main** ([String](#) *args*)

run
public void **run** (*AppConfiguration* *configuration*, Environment *environment*)

4.1.2 uk.ac.ox.it.cha.auth

AppAuthenticator

public class **AppAuthenticator** implements Authenticator<BasicCredentials, *User*>

Author martinfilliau

Fields

apiAuth
ApiAuth **apiAuth**

Constructors

AppAuthenticator
public **AppAuthenticator** (*ApiAuth apiAuth*)

Methods

authenticate
public Optional<*User*> **authenticate** (BasicCredentials *credentials*)

User

public class **User**

> **Author** martinfilliau

Constructors

User
public **User** (String *username*)

Methods

getPassword
public String **getPassword** ( )

getUsername
public String **getUsername** ( )

setPassword
public void **setPassword** (String *password*)

setUsername
public void **setUsername** (String *username*)

4.1.3 uk.ac.ox.it.cha.configuration

ApiAuth

public class **ApiAuth** extends Configuration

> **Author** martinfilliau

Methods

getPassword
public String **getPassword** ( )

getUser
public String **getUser** ( )

AppConfiguration

public class **AppConfiguration** extends Configuration
> Application configuration

> > **Author** martinfilliau

Fields

apiauth
ApiAuth **apiauth**

cucm
CucmConfiguration **cucm**

Methods

getApiauth
public *ApiAuth* **getApiauth** ( )

getCucm
public *CucmConfiguration* **getCucm** ( )

CucmConfiguration

public class **CucmConfiguration**
> Configuration for the CUCM server

> > **Author** martinfilliau

Methods

getHost
public String **getHost** ( )
> Get the host to connect to

getPassword
public String **getPassword** ( )
> Get the password

getUser
public String **getUser** ( )
> Get the username

4.1.4 uk.ac.ox.it.cha.health

CucmAxlServiceHealthCheck

public class **CucmAxlServiceHealthCheck** extends HealthCheck
> Health check to CUCM AXL Service

> > **Author** martinfilliau

Constructors

CucmAxlServiceHealthCheck
public **CucmAxlServiceHealthCheck** (AXLPort *service*)

Methods

check
protected Result **check** ( )
> Make a query to get the OS version

> > **Returns** healthy if there was no exception else unhealthy

4.1.5 uk.ac.ox.it.cha.representations

Phone

public class **Phone**
> Represents a Phone

> > **Author** martinfilliau

Constructors

Phone
public **Phone** ( )

Phone
public **Phone** (GetPhoneRes *gpr*)
> Build a Phone object from a GetPhoneRes object

> > **Parameters**

> > > • **gpr** – GetPhoneRes object

Methods

equals
public boolean **equals** (Object *o*)

getDescription
public String **getDescription**()

getDirns
public List<String> **getDirns**()

getModel
public String **getModel**()

getName
public String **getName**()

getProduct
public String **getProduct**()

getSpeeddials
public List<*Speeddial*> **getSpeeddials**()

getUuid
public String **getUuid**()

hashCode
public int **hashCode**()

setDescription
public void **setDescription**(String *description*)

setDirns
public void **setDirns**(List<String> *dirns*)

setModel
public void **setModel**(String *model*)

setName
public void **setName**(String *name*)

setProduct
public void **setProduct**(String *product*)

setSpeeddials
public void **setSpeeddials**(List<*Speeddial*> *speeddials*)

setUuid
public void **setUuid**(String *uuid*)

toString
public String **toString** ()

Speeddial

public class **Speeddial**

> Represents a Speed dial

> > **Author** martinfilliau

Constructors

Speeddial
public **Speeddial** ()

Speeddial
public **Speeddial** (RSpeeddial *sd*)

> Construct a Speeddial object from CUCM

> > **Parameters**

> > > • **sd** – RSpeeddial object from CUCM

Methods

equals
public boolean **equals** (Object *o*)

getDirn
public String **getDirn** ()

getIndex
public String **getIndex** ()

getLabel
public String **getLabel** ()

getXSpeeddial
public XSpeeddial **getXSpeeddial** ()

> Get an XSpeeddial object (used in queries made to CUCM)

> > **Returns** XSpeeddial object

hashCode
public int **hashCode** ()

setDirn
public void **setDirn** (String *dirn*)

setIndex
public void **setIndex** (String *index*)

setLabel
public void **setLabel** (String *label*)

toString
public String **toString** ( )

4.1.6 uk.ac.ox.it.cha.resources

PhoneResource

public class **PhoneResource**
> Phone resource Expose information about phones

> > **Author** martinfilliau

Constructors

PhoneResource
public **PhoneResource** (AXLPort *axlService*)

Methods

get
public List<*Phone*> **get** (IntParam *dirn*, *User user*)
> Get phone information from a directory number

> > **Parameters**

> > > • **dirn** – directory number

> > > • **user** – authenticated user

> > **Returns** list of phones corresponding to this directory number

SpeeddialsResource

public class **SpeeddialsResource**
> Speed dials resource Allow operations on speed dials

> > **Author** martinfilliau

Constructors

SpeeddialsResource
public **SpeeddialsResource** (AXLPort *axlService*)

Methods

updateSpeeddials

public Response **updateSpeeddials** (String *phone*, List<*Speeddial*> *speeddials*, *User user*)

Update the speed dials for the given phone name

> **Parameters**
>
> > - **phone** – name of the phone to update
> >
> > - **speeddials** – list of Speeddial
> >
> > - **user** – authenticated user
>
> **Returns** response code from the SOAP web service

4.1.7 uk.ac.ox.it.cha.services

CucmAxlService

public class **CucmAxlService** implements Managed

> **Author** martinfilliau

Constructors

CucmAxlService

public **CucmAxlService** (AXLPort *service*)

Methods

start

public void **start** ()

stop

public void **stop** ()

Indices and tables

- genindex
- modindex
- search

/phone

GET /phone/, 3

/speeddials

POST /speeddials, 4

start() (Java method), 14
stop() (Java method), 14

T

toString() (Java method), 12, 13

U

uk.ac.ox.it.cha (package), 7
uk.ac.ox.it.cha.auth (package), 7
uk.ac.ox.it.cha.configuration (package), 8
uk.ac.ox.it.cha.health (package), 10
uk.ac.ox.it.cha.representations (package), 10
uk.ac.ox.it.cha.resources (package), 13
uk.ac.ox.it.cha.services (package), 14
updateSpeeddials(String, List, User) (Java method), 14
User (Java class), 8
User(String) (Java constructor), 8