Alice Protocol

February 23, 2012

Contents

1 Introduction

Alice acts as the connector between one or more NAME_OF_PRODUCT clients and a data store/soft PBX that is comprised of a mixture of LDAP/PostgreSQL servers and one ore more FreeSWITCH servers. Her main task is to get and set data via a HTTP/JSON based protocol. This document is the final authority on the protocol understood by Alice.

2 The Protocol

In its most basic form the protocol consists of a set of "get" and "set" interfaces. You can either get some data from Alice, or you can tell her to set some data.

When data is requested (get) the search parameters are always given as a HTTP GET query string and the result is always returned as a JSON string. When data is added/updated/deleted (set) the parameters are always given as a HTTP POST request and the confirmation returned from Alice will always be a JSON string.

In order to access any of the interfaces, a client must first authenticate itself. This is done using OpenID. The login interface is the only interface that can be accessed without prior login.

All data handled by the protocol is expected to be UTF-8 encoded. Alice does not check whether this is in fact the case, so when implementing clients, be sure to verify that all data sent to Alice is in fact UTF-8 encoded.

All interfaces should of course be prefixed with whatever server:port is used for a given Alice instance.

3 Getting Data

3.1 /get/company?o=[COMPANY]

This interface return all the data associated with COMPANY. If the given company doesn't exist in the database, then an empty JSON string ({}) is returned.

Only company data is returned when calling this interface. No employee data is returned. Calling for example

```
/get/company?o=Hansen%20VVS
```

will return a JSON string similar to this:

```
{
"o=Hansen VVS,dc=example,dc=com":
    {
    "l": "Næstved",
    "o": "Hansen VVS",
    "street": "Slagelsevej 13",
    "objectClass":
        [
        "top",
        "organization"
        ],
        "telephoneNumber": "+45 5544 3321",
        "postalCode": "DK-4700"
        }
    }
```

TODO: Document each JSON field.

3.2 /get/person?o=[COMPANY]&cn=[PERSON]

This interface return the data associated with PERSON employed by COMPANY. Calling for example

```
/get/person?o=Hansen%20VVS&cn=Brian%20Hansen
```

will return a JSON string similar to this:

```
{
"cn=Brian Hansen,o=HansenVVS,dc=example,dc=com":
    {
    "cn": "Brian Hansen",
    "objectClass":
        [
        "person",
        "organizationalPerson",
        "top"
        ],
    "telephoneNumber": "+45 2266 1155",
    "sn": "Hansen",
    "title": "Gasmester"
    }
}
```

TODO: Document each JSON field.

3.3 /get/persons?o=[COMPANY]

This interface return the persons associated with COMPANY. Calling for example

```
/get/persons?o=Hansen%20VVS
```

will return a JSON string similar to this:

```
{
"cn=Ole Johnsen,o=Hansen VVS,dc=example,dc=com":
    {
    "cn": "Ole Kappel Johnsen",
    "objectClass":
        [
            "person",
            "organizationalPerson",
            "top"
            ],
        "telephoneNumber": "+45 55 11 66 35",
        "sn": "Johnsen",
        "title": "Svend"
        },
    "cn=Brian Hansen,o=Hansen VVS,dc=example,dc=com":
        {
        "cn": "Brian Hansen",
        "objectClass":
            [
            "person",
            "organizationalPerson",
            "top"
            ],
        "telephoneNumber": "+45 2266 1155",
        "sn": "Hansen",
        "title": "Gasmester"
        }
    }
```

TODO: Document each JSON field.

3.4 /get/queue

This interface return the current call queue. This is updated once each second, so polling the /get/queue interface more often than once each second is both wasteful and pointless. Calling for example:

```
    /get/queue
```

will return a JSON string like this:

```
    {
    "normal":
        [
        {
        "UTC_start_date": "2012-02-22 14:23:30",
        "id": "GDhcf2VBww",
        "unix_timestamp": "1329920610",
        "callee": "mU1Ff16h",
        "caller": "d7sIp1kR"
        }
        ],
    "high":
        [
    {
    "UTC_start_date": "2012-02-22 14:23:11",
    "id": "bRbYsMUVqx",
    "unix_timestamp": "1329920591",
    "callee": "pMT3k2fb",
    "caller": "oCgDF7ua"
    }
    ],
"low":
    [
    {
    "UTC_start_date": "2012-02-22 14:23:17",             "id": "VV8BFqGpqG",
    "unix_timestamp": "1329920597",
    "callee": "7ar4Y1qc",            "caller":
    "ZQsRogwB"
    }
    ]
}
```

TODO: Document each JSON field.

3.5 /get/queue?kind=length

This interface returns the length of the current call queue. Calling

```
/get/queue?kind=length
```

will return a JSON string like this:

```
{
"length":"7"
}
```

TODO: Document each JSON field.