Verto Communicator

About

Verto Communicator is one example of web clients that can be implemented on top of mod_verto.

Dependencies

We'll use NodeJS based tools to be able to quickly run tests, lint, build and serve Verto Communicator.

Installing node and npm on a Mac could simple as running:

```
brew install npm
```

Or you can follow the instructions on NodeJS site.

The tools we'll need for now are:

```
npm install -g grunt grunt-cli bower
```

You can test Verto Communicator features via dialing "3500" number on server with vanilla config.

FreeSWITCH

Specific settings

livearray-json-status

You'll need to set flag livearray-json-status in your members-flags, for more info check mod_conference page.

livearray-json-status flag in your conference profile

```
<param name="conference-flags" value="video-floor-only|rfc-4579|livearray-sync|minimize-video-encoding|livearray-json-status"/>
```

caller-controls

Edit you conference.conf.xml and make sure you have these caller-controls:

caller-controls for Verto Communicator

```
<caller-controls>
  <group name="default">
    <control action="deaf mute" digits="**"/>
<control action="mute" digits="0"/>
<control action="vmute" digits="*0"/>
<control action="vmute snap" digits="*1"/>
<control action="vmute snapoff" digits="*2"/>
<control action="energy up" digits="9"/>
<control action="energy equ" digits="8"/>
 <control action="energy dn" digits="7"/>
<control action="vol talk up" digits="3"/>
<control action="vol talk zero" digits="2"/>
<control action="vol talk dn" digits="1"/>
<control action="vol listen up" digits="6"/>
<control action="vol listen zero" digits="5"/>
<control action="vol listen dn" digits="4"/>
<control action="hangup" digits="#"/>
</group>
</caller-controls>
```

Building for Production

After installing the dependencies, simply build it using:

```
cd verto_communicator
npm install
bower install
grunt build
```

The commands above with leave a *dist/* folder on the currrent working directory with all the files necessary to deploy Verto Communicator minified. Just copy them over to any webserver and that's it.

Developing and Contributing

After installing the dependencies, let's setup the project:

```
cd verto_communicator
npm install
bower install
grunt serve
```

This will leave a server running on your local machine serving the necessary files to Verto Communicator to work. Just open Chrome and browse to: https://localhost:9001.

Languages

Verto Communicator now has i18n support. Currently we support:

1. English - default
2. Italy

The language detection is based on the browser language.

Adding translation

To add more languages you need to:

```
cd /usr/src/freeswitch/html5/verto/verto_communicator/src/
cp locales/locale-en.json locales/locale-yourlanguage.json
```

Translate the strings and register the new available language:

html5/verto/verto_communicator/src/vertoApp/vertoApp.module.js

```
$translateProvider
      .useStaticFilesLoader({
        prefix: 'locales/locale-',
        suffix: '.json'
      })
      .registerAvailableLanguageKeys(['en', 'it', 'pt', 'fr'], {
        'en': 'en',
        'en_GB': 'en',
        'en_US': 'en',
        'it': 'it',
        'it_IT': 'it',
                    'fr': 'fr',
                    'fr_FR': 'fr',
        'fr_CA': 'fr',
        'pt': 'pt',
        'pt_BR': 'pt',
        'pt_PT': 'pt'
      })
      .preferredLanguage('en')
      .fallbackLanguage('en')
      .determinePreferredLanguage()
      .useSanitizeValueStrategy(null);
```

Add your language in the registerAvailableLanguageKeys function. All done.

See Also

- mod_verto