Tasker Application Server

The Tasker Application Server runs on Node and interfaces with the Oracle Database server set up in Chapter 2.

System Requirements

Technically, the application server can run on Windows, Linux, or Mac OS X. As long as an Oracle Instant Client is available for your platform and you can install Node, the application server should work.

Prepare your system

Although you can run Tasker under any user, on a Linux system, it's best to create a new user for the node process to run in:

1. As root, add the `taskeradmin` group:

```
groupadd taskeradmin
```

2. Add the `tasker` user:

```
useradd -G taskeradmin tasker
```

3. Assign a password to `tasker`:

```
passwd tasker
```

4. Allow `tasker` to `sudo`:

   a. `visudo`

   b. Find the line that looks like this: `root ALL=(ALL) ALL` and add a new line below it:

```
    tasker    ALL=(ALL)       ALL
```

Download and Install Node

If you're running Windows or Mac OS X, you can download and install Node from http://nodejs.org. Follow the steps as presented by the installer to install Node.

For Linux, do the following: (Based on part from https://www.digitalocean.com/community/tutorials/how-to-compile-node-js-with-npm-from-source-on-centos-6)

1. Make sure you have the following pre-requisites:

```
yum install man bc xauth wget gcc gcc-c++ automake autoconf libtoolize make git
```

2. Change to the `/opt` directory and download the latest version of Node:

```
wget http://nodejs.org/dist/node-latest.tar.gz
```

3. Untar the file:

```
tar xzvf node-latest.tar.gz
```

4. Change to the newly extracted directory

```
cd node-v*
```

5. Configure:

```
./configure
```

6. Build from source; this will probably take awhile

```
make
```

7. When done, install:

```
make install
```

8. Verify node is installed:

```
node --version
npm --version
```

9. In order to use `node` or `npm` with `sudo`, execute `visudo` and add `/usr/local/bin` to the `Defaults secure_path=...`
10. Install `n`:

```
npm install -g n
```

11. To update, you can use

```
sudo n stable
```

Install Oracle Instant Client

In order for us to use Node.js and Oracle, we need to install the Oracle Instant Client. The following is based on instructions at https://github.com/joeferner/node-oracle/blob/master/INSTALL.md, and assumes a Linux operating system.

- Download the Oracle Instant Client for your machine from http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html. Download either the Basic or Basic Lite package as well as the SDK package. Ensure that you select the proper architecture for your system.
- As `tasker`:

```
sudo rpm -ivh oracle-instantclient12.1-basic-12.1.0.1.0-1.x86_64.rpm
sudo rpm -ivh oracle-instantclient12.1-devel-12.1.0.1.0-1.x86_64.rpm
```

- Link libraries (may already exist) (in `/usr/lib/oracle/12.1/client64/lib`):

```
ln -s libclntsh.so.12.1 libclntsh.so
ln -s libocci.so.12.1 libocci.so
```

- Add the shared objects to the ld cache:

```
echo '/usr/lib/oracle/12.1/client64/lib/' | sudo tee -a /etc/ld.so.conf.d/oracle_instant_client
sudo ldconfig
```

- Set up `.bash_profile` for `tasker` to point to Oracle Instant Client:

```
export OCI_HOME=/usr/lib/oracle/12.1/client64/lib
export OCI_LIB_DIR=$OCI_HOME
export OCI_INCLUDE_DIR=$OCI_HOME/sdk/include
export OCI_VERSION=12
export NLS_LANG=AMERICAN_AMERICA.UTF8
```

> Note that your installation may put the `include` files in `/usr/include` instead. If so, use the following `export`:
>
> ```
> export OCI_INCLUDE_DIR=/usr/include/oracle/12.1/client64
> ```

Install Tasker

> Note that these steps clone the *entire* PhoneGap Enterprise repository. If this isn't what you want, you'll need to take other steps to clone only the `tasker-srv` directory.

- Create a `tasker` directory:

```
mkdir tasker
```

- Clone the repository using Git:

```
git clone https://github.com/kerrishotts/PhoneGap-Enterprise-Code-Bundle
```

- Navigate to the `tasker-srv` directory:

```
cd PhoneGap-Enterprise-Code-Bundle/tasker-srv
```

- Install dependencies (note: this will fail if the Oracle instant Client is not installed correctly):

```
npm install
```

- Add your certificates to `_certs`. If they aren't named `ssca.cer` and `tasker1.cer`, you will need to edit the `tasker-srv/config/development.json` or `tasker-srv/config.production.json` appropriately.

- Make sure the configuration files in `config` have the correct login information for your database.

- Start the server:

```
export NODE_ENV=development; npm start
```

> Note: if you want to use a different port from `4443`, modify the configuration files appropriately.

Tasker and Self-Signed Certificates

If you must use self-signed certificates, you should perform the following steps (from http://blog.httpwatch.-com/2013/12/12/five-tips-for-using-self-signed-ssl-certificates-with-ios/):

1. Create your own certificate authority and root CA certificate as follows:

```
openssl genrsa -out ssca.key 2048
openssl req -x509 -new -key ssca.key -out ssca.cer -days 730 -subj /CN="Example Corporation CA"
```

2. Install `ssca.cer` on your devices (you can e-mail them to the device, or you can use the platform's management software to push a profile containing the certificate to your users).

3. Create your self-signed certificate(s) as follows (replacing domains as appropriate):

```
openssl genrsa -out tasker1.key 2048
openssl req -new -out tasker1.req -key tasker1.key -subj /CN=pge-as.photokandy.com
openssl x509 -req -in tasker1.req -out tasker1.cer -CAkey ssca.key -CA ssca.cer -days 365 -CAcreate
```

4. Install your self-signed certificates on your mobile device.

5. Copy the certs you've created to the `_certs` directory where you installed the Tasker server. If the filenames are different, you'll need to properly update the configuration files as mentioned in in the prior section.

Verify Tasker Server

To verify that the Tasker Server is working, just access the server in a browser. For example, if your server is running on `pge-as.photokandy.com:4443`, just navigate to https://pge-as.photokandy.com:4443 in your browser. You should receive formatted API documentation in response.