Testing on Headless CI Systems (Travis CI, Jenkins)

Being based on Chromium, Electron requires a display driver to function. If Chromium can't find a display driver, Electron will fail to launch - and therefore not executing any of your tests, regardless of how you are running them. Testing Electron-based apps on Travis, CircleCI, Jenkins or similar Systems requires therefore a little bit of configuration. In essence, we need to use a virtual display driver.

Configuring the Virtual Display Server

First, install Xvfb. It's a virtual framebuffer, implementing the X11 display server protocol - it performs all graphical operations in memory without showing any screen output, which is exactly what we need.

Then, create a virtual Xvfb screen and export an environment variable called DISPLAY that points to it. Chromium in Electron will automatically look for `$DISPLAY`, so no further configuration of your app is required. This step can be automated with Anaïs Betts' xvfb-maybe: Prepend your test commands with `xvfb-maybe` and the little tool will automatically configure Xvfb, if required by the current system. On Windows or macOS, it will do nothing.

```
## On Windows or macOS, this invokes electron-mocha
## On Linux, if we are in a headless environment, this will be equivalent
## to xvfb-run electron-mocha ./test/*.js
xvfb-maybe electron-mocha ./test/*.js
```

Travis CI

On Travis, your `.travis.yml` should look roughly like this:

```
addons:
  apt:
    packages:
      - xvfb

install:
  - export DISPLAY=':99.0'
  - Xvfb :99 -screen 0 1024x768x24 > /dev/null 2>&1 &
```

Jenkins

For Jenkins, a Xvfb plugin is available.

CircleCI

CircleCI is awesome and has Xvfb and `$DISPLAY` already set up, so no further configuration is required.

AppVeyor

AppVeyor runs on Windows, supporting Selenium, Chromium, Electron and similar tools out of the box - no configuration is required.