Base Distribution - Task # 48241

Work Package # 48367 (New): [WIP] End-To-End Testing of Neos with Behat

Status:	New	**Priority:**	Should have
Author:	Christopher Hlubek	**Category:**	
Created:	2013-05-16	**Assigned To:**	Robert Lemke
Updated:	2013-05-21	**Due date:**	
Subject:	Create sandboxed Jenkins testing environment		

Description

Running tests (pre-merge / post-merge) should be separated from release management and deployment.

Some notes about the additional setup for running Behat tests with headless Selenium:

CI Setup for Selenium

- Install Xvfb (aptitude install xvfb)
- Download init script from https://gist.github.com/jterrace/2911875 to /etc/init.d/xvfb
- chmod +x /etc/init.d/xvfb
- sudo update-rc.d xvfb defaults

- Install Firefox

- Install Selenium server into /usr/local/lib/selenium:
  - Download http://selenium.googlecode.com/files/selenium-server-standalone-2.32.0.jar
- Download init script from http://www.danstraw.com/blog/wp-content/uploads/2010/11/selenium.txt
- Added 'export DISPLAY=":1.0"' before starting Selenium server to set the display for Xvfb

- Run Apache on 8081 (parallel to Nginx)
- Create VHost in Apache
- Install MySQL

Ideas:

- Post only message on failure (instead of -1)
- Start Xvfb using Jenkins Xvfb plugin (run in parallel in start / stop after tests)

- Capture video (headless gem)
  - Start capturing video before each scenario

History

#1 - 2013-05-17 18:40 - Sebastian Kurfuerst

- Parent task set to #48367

#2 - 2013-05-21 10:52 - Christopher Hlubek

We created an account on Sauce Labs (http://saucelabs.com/) for Neos (it's free for Open Source projects). They provide excellent browser testing VMs and support for Selenium WebDriver, so we can implement a build matrix for the Behat tests on different device / browser combinations. It's also much easier to debug errors or problems in failing tests since they provide a detailed log and captured video for each test.

I already installed a Jenkins plugin for Sauce Labs, but we still have to split the job into at least one downstream job for running the Behat tests. A build

pipeline for review could look like this:

- Gerrit Trigger for Neos package dependencies (on patchset created)
  - Run Unit Tests -> publish results
  - Run Functional Tests -> publish results
  - Run Behat tests on different browsers (if Unit / Functional tests are okay)
  - Publish final results to Gerrit (Set verified +1, but do not set -1 for failure)

- Gerrit Trigger for Neos package dependencies (on patchset created)
  - Run Unit Tests -> publish results
  - Run Functional Tests -> publish results