

Development

Indholdsfortegnelse

Hermed lille writeup der beskriver hvorledes rettelser skal komme fra et Dev miljø til prod:.....	1
Indlæggelse af nyt modul.....	1
Indlæggelse af theme rettelse.....	1
Indlæggelse af konfiguration ændring / Struktur.....	1
GIT Flow.....	2
Hurtige Guidelines:.....	2

Hermed lille writeup der beskriver hvorledes rettelser skal komme fra et Dev miljø til prod:

Primær udvikling sker i github "OS2Web":<http://www.github.com/OS2web/os2web> repository. Her kan commites custom moduler og features.

Theme udvikling sker i github "os2web-theme":<http://www.github.com/OS2web/os2web-theme> repository.

Deployment scripts findes i github "OS2Web-deploy":<http://www.github.com/OS2web/os2web-deploy> repository. Ændringer her skal gøres med omtanke og make filer heri er ikke til alm. udvikling.

Guide til git kan findes "her":<http://help.github.com/set-up-git-redirect>.

Indlæggelse af nyt modul

- Moduler skal tilføjes til makefilen (*os2web.make*). Syntax kan ses i eksisterende tilføjelser.
- Makefilen commits og pushes og et reroll køres til kontrol. Dette gentages på en staging maskine før det køres på prod.
- Patches til moduler inddeles i to kategorier og håndteres forskelligt:
 - Bugfixes til eller fra et moduls issue queue, dvs. en generel rettelse, skal linkes online til patch i det tilhørende issue.
 - Rettelser der er en tilpasning til et contrib modul, der kun er relevant i OS2 regi ligges i patches folderen og linkes lokalt.

Indlæggelse af theme rettelse

- Theme rettelser pushes til theme repository. et pull på et staging site vil bekræft ønsket funktion, og et pull eller kan derefter udføres på prod.

Indlæggelse af konfiguration ændring / Struktur

- Til dette bruges "Features":<http://drupal.org/project/features> modulet.

- Struktur er bl.a.:
 - Views
 - Panels
 - Contexts
 - Deltablocks
- Konfiguration er elementer i *variables* tabel, der typisk sættes i en konfigurations side.
- Ændringen enten tilføjes til en eksisterende features eller der skal oprettes en ny.
 - Featuren oprettes enten på prod eller lokalt vha. recreate funktionen
 - Featuren uploades til en staging server og revertes, her kontrollere man at alt er med og korrekt
 - Features committes til git og kommer med i næste reroll af sitet.

GIT Flow

Git Flow fungerer kun på det lokale repository, og instillingerne kan derfor ikke smides op i github. Det er derfor nødvendigt med en frisk opsætning af gitflow til de repositories man skal arbejde i:

Hent Git-flow via vejledning fra github: <https://github.com/nvie/gitflow/> Direkte link <https://github.com/nvie/gitflow/wiki/Installation>

Hvis du kører Ubuntu / Debian brug "apt-get install git-flow"

```
$ cd <git repo rod f.eks. modules/os2web>
$ git flow init`
Which branch should be used for bringing forth production releases?
- dev
Branch name for production releases: [] *master*
Which branch should be used for integration of the "next release"?
- dev
Branch name for "next release" development: [develop] *dev*
How to name your supporting branch prefixes?
Feature branches? [feature/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? [] *v*
```

Derefter kan git flow bruges som nævnt ovenfor

Hurtige Guidelines:

- Al udvikling af moduler/features sker i feature branches. f.eks. navngivet efter relevant story
- Alle Hotfixes *skal* laves vha. hotfix branches da de ellers ikke kommer ud på prod
- Server opsætning:
 - Produktion: master
 - Staging: dev
 - Testing: dev eller div. feature banches