

This document describes following topics:

[Releasing from local machine](#)

[Releasing skin to local environment](#)

[Releasing skin to staging environment](#)

[Releasing skin to live environment](#)

[Required files](#)

[Releasing via Jenkins](#)

Releasing from local machine

In following 3 sections all commands/steps are performed on developer's local machine. We're also using Jenkins, which is described in last section.

Releasing skin to **local** environment

WEB team uses grunt for deployment.
Built sources are kept in **build/** folder.
Grunt file is here **build/Gruntfile.js**.

In order to deploy project locally on your machine, perform the steps:

- `cd build/`
- `npm install` (run just once, to install all dependencies)
- `grunt [dev] [--skin=marsbet.com]`
 - **dev** - skips stripping `console.log()`s from source, making build process much faster
 - if run without `--skin` option, it will build `vbet.com` by default
- you'll find build sources in `build/app/marsbet.com/` folder, they are ready to be put on production server (if run without `dev` option of course)
- in order to test locally this build version, just open [http://\[your_local_host\]/build/app/marsbet.com/](http://[your_local_host]/build/app/marsbet.com/)

Releasing skin to **staging** environment

- `cd build/`
- `grunt build-and-deploy-all --skin=kickoff102bet.com`
`--skin-config=kickoff102bet.com`
`--deploy-path=/var/www/html/staging/Kickoff102bet/`
`--deploy-key=staging --deploy-host=wordpress.betconstruct.int`
 - **build-and-deploy-all** - this tasks does everything, there are various other options (only css, fonts, ...). For

- that list please go to build/Gruntfile.js and search for registerTask('deploy
- o **--skin** - matches skin folder name under /skins
 - o **--skin-config** - optional, is necessary only if skin name is different from its config
 - o **--deploy-path** - this a path in staging server where built sources will be uploaded to. If not provided will be read from servers.json file (described below)
 - o **--deploy-key** - is used to match/get username, password for an environment kept in build/.ftppass file
 - o **--deploy-host** - host name to upload built sources to. If not provided will be read from servers.json file (described below)
- release will be accessible from following URL:
<http://staging.betconstruct.int/Kickoff102bet.com/#/>

Releasing skin to live environment

Same as for staging environment.

e.g. to release something experimental to vbet.com, under different folder run:

```
grunt deploy-all --skin=vbet.com
--build-and-deploy-path=/var/www/html/ice2014.vbet.com/experiment-
leftMenuHover/
```

Required files

You'll need **build/.ftppass**, and **build/servers.json** files in your project folder to release to stage and live environments. build/.ftppass file is a simple JSON structure with username password for each machine, identified by name e.g.:

```
"staging": {
  "username": "???",
  "password": "???"
},
"betconstruct.me": {
  "username": "???",
  "password": "???"
}
```

Field names here ('staging', 'betconstruct.me', ...) are matching to corresponding values in build/servers.json file:

- 'paths' - holds absolute paths to stage, live machines per skin

- 'hosts' - hosting machine URIs
- 'keys' - is a map where values are referring to keys in build/.ftppass file
 - e.g. to know uname, pass for "vbet.com.staging": "staging" pair, look for "staging" field in build/.ftppass file

Releasing via Jenkins

URL - <http://web.betconstruct.int:8080/>

"[deploy_staging](#)" - staging release plan (uses develop branch)

"[deploy_skin_live_from_git](#)" - live release plan (uses master branch)