



BaseApp
Wydanie 1.0

Jacek Siciarek

December 30 2015

1	Spis treści	1
1.1	Set up	1
1.2	Synchronisation of a code with <i>rsync</i>	3
1.3	Advanced GIT usage	4

Spis treści

1.1 Set up

Here is the sample documentation chapter.

1.1.1 Sphinx update

```
sudo apt-get install texlive-latex-base
sudo apt-get install texlive-latex-recommended
sudo apt-get install texlive-fonts-recommended
sudo apt-get install texlive-latex-extra
sudo apt-get install texlive-lang-polish
sudo apt-get install dvipng

sudo apt-get install xzdec
tlmgr init-usertree
tlmgr install titlesec
tlmgr install framed
tlmgr install threeparttable
tlmgr install wrapfig
tlmgr install upquote
tlmgr install multirow
```

1.1.2 Code block

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char** argv) {

    printf("Hello, World!\n");

    return EXIT_SUCCESS;
}
```

1.1.3 Mathematical formula

$$\int_a^b u \frac{d^2v}{dx^2} dx = u \frac{dv}{dx} \Big|_a^b - \int_a^b \frac{du}{dx} \frac{dv}{dx} dx.$$

can be created with the following:

```
.. math::

\int_a^bu\frac{d^2v}{dx^2}\,,dx
=\left.u\frac{dv}{dx}\right|_a^b
-\int_a^b\frac{du}{dx}\frac{dv}{dx}\,,dx.
```

1.1.4 Set up Cloud9 environment

```
$ sudo vim /etc/apache2/sites-enabled/001-cloud9.conf
$ apachectl restart
```

1.1.5 Useful commands

Apache commands

```
$ apachectl start
$ apachectl restart
$ apachectl stop
```

MySQL Database console access

```
$ mysql-ctl cli
```

PostgreSQL Database console access

```
$ sudo sudo -u postgres psql
```

```
CREATE USER root SUPERUSER PASSWORD 'pass';
```

MySQL server commands

```
$ mysql-ctl start
```

PostgreSQL server commands

```
$ sudo service postgresql start
```

1.1.6 Deploy Symfony Application

- <http://symfony.com/doc/current/cookbook/deployment/tools.html>

1.2 Synchronisation of a code with *rsync*

Here is the description of synchronizing local code with remote server using *rsync*.

1.2.1 Excluded files list

You can place following list in the file `/app/config/rsync_exclude.txt`

```
*~
*bak
.git*
/go
app/data.db3
app/bootstrap.php.cache
app/cache
app/logs
app/config/parameters.yml
app/config/rsync_exclude.txt
features
nbproject
.idea
php.ini
bin
temp
behat.*
src/Application/MainBundle/Tests
src/Application/MainBundle/Resources/doc
build.xml
composer.json
composer.lock
composer.phar
LICENSE
properties.conf
properties.conf.dist
README.md
sync
go
vendor
web
```

1.2.2 Executable script

You can place the script in main project directory eg. `./sync`

```
#!/usr/bin/env bash

PRIVATE_KEY=~/.ssh/maxie.id_rsa
```

```

REMOTE_USER=maxie
REMOTE_HOST=maxiehill.com
REMOTE_PORT=22
REMOTE_DIR=/home/maxie/workspace/service.maxiehill.com

SOURCE=.
TARGET=$REMOTE_USER@$REMOTE_HOST:$REMOTE_DIR

MODE=--dry-run

if [ "$1" = "--go" ]
then
    MODE=
fi

clear

rsync \
$MODE \
--rsh "ssh -p$REMOTE_PORT -i $PRIVATE_KEY" \
--itemize-changes \
--verbose \
--human-readable \
--times \
--progress \
--links \
--stats \
--compress \
--recursive \
--links \
--delete \
--exclude-from=app/config/rsync_exclude.txt \
$SOURCE $TARGET

if [ "$1" = "--go" ]
then
    ssh -p $REMOTE_PORT -i $PRIVATE_KEY $REMOTE_USER@$REMOTE_HOST "cd $REMOTE_DI
fi

```

1.3 Advanced GIT usage

1.3.1 Show git repo history

rich

```
$ git log
```

brief

```
$ git log --pretty=oneline
```

1.3.2 Revert changes in a single file

```
$ git reset HEAD /your/path/to/the/project/Sender.php  
$ git checkout -- /your/path/to/the/project/Sender.php
```

1.3.3 Git tags support

Let us assume that our release has name “v1.0.0”

Create tag

```
$ git tag -a v1.0.0 -m "The very first release of the Application."
```

List tags

```
$ git tag
```

Remove tag

local

```
$ git tag -d v1.0.1
```

remote

```
$ git push origin :refs/tags/v1.0.1
```

1.3.4 Git branches support

Clone master

```
$ git clone git@git.enovatis.pl:ino  
$ git clone git@git.enovatis.pl:ino.git  
$ git clone git@git.enovatis.pl:ino ino  
$ git clone git@git.enovatis.pl:ino.git ino  
$ git clone -b master git@git.enovatis.pl:ino  
$ git clone -b master git@git.enovatis.pl:ino.git  
$ git clone -b master git@git.enovatis.pl:ino ino  
$ git clone -b master git@git.enovatis.pl:ino.git ino
```

Clone branch

```
$ git clone -b devel git@git.enovatis.pl:ino
```

Let us assume that branch is named “devel”.

Create branch

Run single command

```
$ git checkout -b devel
```

or two

```
$ git branch devel  
$ git checkout devel
```

then send it to remote server

```
$ git push --set-upstream origin devel
```

Delete branch

local

```
$ git branch -d devel
```

remote

```
$ git push origin --delete devel
```

Show branches

```
$ git branch
```

Switch to specific branch

```
$ git checkout devel  
$ git checkout master
```

Show diff of previous version

```
$ git diff HEAD@{1}
```


Show diff between branches

Implicit diff

```
$ git diff master..devel
```

Name and status

```
$ git diff --name-status master..devel
```

More info

```
$ git diff --stat master..devel
```

Merge branch

local

```
$ git checkout master  
$ git merge devel
```

remote

```
$ git checkout -b master  
$ git checkout master  
$ git fetch origin  
$ git merge origin/devel
```

Undo merge branch

```
$ git reset --hard origin/master
```

Patch branch

```
$ git diff --no-prefix master..devel > diff.patch  
$ patch < diff.patch
```

Access to branches

<https://www.kernel.org/pub/software/scm/git/docs/howto/update-hook-example.txt>