



Web Production Pipeline

Syncing localhost with live server

What we need?

- Version Control System (we'll be using Mercurial)
- Localhost (I'll be using MAMP)
- Live Server (I'll be using Hostgator)

Background...

Because our goal is to make an easy way to sync our localhost and our live server we need to be able to work on both ends. We need to be able to work on our localhost and have the changes sync to our live server. We also need to be able to work on our live server and have the changes show up on the localhost. So we need to understand the differences between the live server and localhost when it comes to how we setup WordPress.

Static Files - WordPress puts the database server information in the wp-config.php file
Database - WordPress puts the site and homepage URL in the wp-options table.

This creates an issue with knowing whether you're on a localhost or on a different server. So we will have to create some code that will detect whether we are on the localhost or the live server so we can use the same wp-config.php file. And we'll use the power of version control to match the database to the right server.

Version Control

There are many different version control options; CVS, SVN, Git... but for our purposes we're going to use Mercurial. Both Git and Mercurial are very similar.

CVS - <http://cvs.nongnu.org/> (An older version control system)

SVN - <http://subversion.apache.org/> (A solid version control system from the Apache project with lots of GUI apps to take advantage of SVN.)

Git - <http://git-scm.com/> (A very popular version control system. Especially popular with web developers.)

Mercurial - <http://mercurial.selenic.com/> (An easy and popular version control system.)

Setting up Mercurial on a Shared Host on HostGator:

<http://blog.tlensing.org/2010/04/12/installing-mercurial-on-a-shared-web-server-without-root-access-hosteurope/>

STEP #1 - Setup a localhost of WordPress

Follow the process of setting up a localhost installation of WordPress. WebDesign.com has a few webinar walking you through this process.

We need to “activate” version control on this WordPress directory.

Using your Terminal navigate to the localhost folder... in our example /Desktop/testsite/

```
cd /testsite  
hg init  
hg add  
hg commit
```

Now we need to deal with the differences between our localhost and our live server.

Localhost has the following settings in the *wp-config.php*:

```
define('DB_NAME', 'testsite');  
define('DB_USER', 'root');  
define('DB_PASSWORD', '1234');  
define('DB_HOST', 'localhost');
```

The Live Server has the following settings in the *wp-config.php*:

```
define('DB_NAME', 'wpchefc_test');  
define('DB_USER', 'wpchefc_testu');  
define('DB_PASSWORD', '1234qwer');  
define('DB_HOST', 'localhost');
```

So what we need to do is modify the *wp-config.php* file so that it can distinguish whether or not the server or the localhost is being used.

NEW *wp-config.php* file:

```
/* Standard Variables */
$hostname = 'localhost';

/* Check the server */
if ($_SERVER["HTTP_HOST"] === 'wpchef.com') {
    $dbname = 'wpchefc_test';
    $dbuser = 'wpchefc_testu';
    $password = '1234qwer';
} else if ($_SERVER["HTTP_HOST"] === 'localhost') {
    $dbname = 'test';
    $dbuser = 'root';
    $password = '1234';
}

define('DB_NAME', $dbname);

define('DB_USER', $dbuser);

define('DB_PASSWORD', $password);

define('DB_HOST', $hostname);

---
```

Hopefully you can see that this can be modified to show BOTH *wp-config.php* settings.

Now check and make sure it works for your localhost site.

STEP #2 - Using BitBucket.org

Now we will use BitBucket.org as a “middle-man”. We need to create a new repository.

Create new repository
Start from scratch.

Name *(required)*
TestPress Private

Repository type
 Git
 Mercurial

Project management
 Issue tracking
 Wiki

Language
PHP

Description

Website

Create repository

Using SSH from our local computer we will push the local repository up to BitBucket. We do this by the following command. (Make sure you replace it with your own login information)

hg push <https://username@bitbucket.org/username/repository>

Now, we can go over to our live server and clone the BitBucket repository.

hg clone <https://username@bitbucket.org/username/repository>

hg update

This is faster than FTP, because now Mercurial only pushes just the files that need to be changed.

Use BitBucket as your up-to-date version of all your stuff.

Always pull from BitBucket and push to BitBucket.

But what about the database???

We do need to have a folder called **/db/** on both the localhost and the server. The exported database file will be saved there.

We need to setup some scripts!

Pushing to BitBucket

Script: pushbits.bat (Windows) | pushbit.sh (Mac/Server)

```
mysqldump -u username -p password databasename > db/db_sync.sql
hg add db/db_sync.sql
hg commit
hg push https://wpstudio@bitbucket.org/wpstudio/repository
```

Pulling from BitBucket

Script: pullbits.bat (Windows) | pullbits.sh (Mac/Server)

```
hg pull https://wpstudio@bitbucket.org/wpstudio/repository
hg update
cd db
mysql -u username -p password databasename < db_sync.sql
mysql -u username -p password databasename < db.sql
```

There will be a special *db.sql* file sitting in our server that will update the database settings.

*** db.sql SAMPLE***

```
USE databasename
UPDATE wp_options SET option_value="http://localhost/testsite/" WHERE
option_name="siteurl";
UPDATE wp_options SET option_value="http://localhost/testsite" WHERE
option_name="home";
```