

Saturday, July 19. 2008

patError 1.2.0 released

After a year of inactivity in our blog, there are finally some news worth mentioning. Some minutes ago, I release version 1.2.0 of patError. Although this release contains no new features, it might be interesting for any of you using patError, as I adjusted the code to PHP 5. When using this new version, you will get not E_NOTICE or E_STRICT warnings any more as it has been the case with previous releases. There will also be no E_DEPRECATED errors, when you make the switch to PHP 5.3.0. As usual, you can get the release from our PEAR channel.

I decided to port patError to PHP 5 while writing the second edition of my PHP Design Patterns book, which covers patError as a dependency of patTemplate.

Posted by schst in PAT at 19:32

Wednesday, June 20, 2007

Testing FIT

I just uploaded the package Testing_FIT to PEAR. Testing_FIT is the PHP and pearish version of Ward Cunningham's FIT <http://fit.c2.com/>. As such it provides an easy to use framework for acceptance software tests.

```
pear install Testing_FIT-beta
```

Simply install the PEAR package Testing_FIT browse the examples and write your own test classes. Of course, the PHP version of FIT provides the four basic fixtures: Action-, Column-, Row- and Primitive-Fixture. Also there is a Summary-Fixture that shows some statistics.

You'll see, writing tests with this PEAR package is a piece of cake. Besides, the basic idea of FIT - using HTML documents providing test data will convince you.

Posted by gERD in PHP, PAT at 21:20

Monday, June 18, 2007

I18n and such

In lack of a better idea, I founded a new PAT Package: patI18n. So far this package is just a draft and only exists in the SVN repository. As you might guess, patI18n is an abstraction layer for translating issues. Therefore it provides nearly the same interface as the famous GNU Gettext. Still, the actual translation is done by a chain of modules - this way everything is possible and the design stays open to new ideas.

Of course, I didn't start patI18n out of the blue. Last year, I branched patForms to use Gettext for translation instead of the proprietary internal solution. Since then, we have to maintain two branches of patForms. Yesterday I mended the implementation of patForms' gettext branch to use patI18n. This way patForms got rid of all I18n functions (like setLocale) and stuff. My first tests show that everything works fine - still this code is experimental. However you are invited to check out the new versions and give it a go.

Download the sources from our subversion server:

patI18n: `svn co http://www.php-tools.net/svn/patI18n/trunk/ patI18n`

patForms: `svn co http://www.php-tools.net/svn/patForms/branches/gettext/ patForms`

View the source code of patI18n at gERD's playground.

Posted by gERD in PHP, PAT at 08:45

Saturday, June 2, 2007

patTemplate 3.1.0 stable released

After two years of development we finally released patTemplate 3.1.0 as a stable version. The changes since the last beta release are minimal. You can read about all changes since the last stable release in the changelog. patTemplate can be downloaded from our PEAR channel server or easily installed via the PEAR installer:

```
$ pear channel-discover pear.php-tools.net
```

```
$ pear install pat/patTemplate
```

More information can be found the the patTemplate development site.

Posted by schst in PHP, PAT at 17:30

Thursday, February 1, 2007

patTemplate Documentation in Bahasa Indonesia

We are vary proud to hear that there are patTemplate enthusiasts around the world. Actually, we just learned that there is a patTemplate Documentation in Bahasa Indonesia available. Thanks a lot for translation!

Posted by gERD in PHP, PAT at 20:41

Tuesday, October 17, 2006

FIT for PHP

As I'm going to talk about FIT at the PHP Conference, I like to invite you to check out the preview of PHPFIT package. Simply download and install the PEAR package or tar archive.

FIT stands for Framework for Integrated Test (See: FIT Wiki) realizes acceptance tests with heavy use of HTML documents. Those documents do not just contain the usual blah-blah but hard facts. The facts are actually stored in (HTML) tables. This is where FIT - respectively PHPFIT - comes in. The FIT knows which testing class must be executed for a particular table and which function to call to calculate the results of a specific cell and the framework makes it very easy for you to implement concrete test-classes.

Also FIT simply handles HTML documents. As such they can be viewed in common browsers and even created with Word-a-like editors. Hence FIT goes far beyond what traditional unit testing offers (FIT is not meant to replace unit-tests) and may enrich your next project.

Hopefully PHPFIT will be available as official PEAR package. Unfortunately I lost my password. So I was not capable to push the proposal any further. But this will change.

Posted by gERD in PHP, PAT at 21:34

Blog Export: a programmer's best friend, <http://blog.php-tools.net/>

Tuesday, September 5, 2006

XJConf for PHP

Today, Frank and I released the first version of XJConf For PHP. This is a port of the original XJConf I developed for Java. XJConf allows you to map complex XML documents to data structures by defining the mapping rules in a simple XML-based language. You may define the type for each tag (objects, arrays, primitives) as well as how its attributes, child elements and enclosed character data should be treated. When creating objects, you may also define the setter methods to nest the objects. There are plenty of examples available in the documentation of the original java version. Most of the examples have been ported to PHP as well by Frank, who did most of the work in this project.

If you want to give XJConf For PHP a try, you can get it via our PEAR channel or from SVN at <http://svn.xjconf.net/XJConfForPHP/>.

The fantastic XJConf logo has been developed and designed by Georg Rothweiler.

Posted by schst in PHP, PAT at 21:56

Sunday, August 20, 2006

New releases of patTemplate and patForms

After I finally finished my book (and my contribution to Exploring PHP) I finally have enough time to work on all of my open source projects again and today made two new releases. patTemplate 3.1.0b1 contains tons of changes since the last stable version. Most of the features that users have been requesting for years have been incorporated into this release, like the possibility to read templates from a database: \$tmpl = new patTemplate();

```
$tmpl->setRoot('mysql://root:@localhost/test', 'DB');
```

```
$tmpl->readTemplatesFromInput("SELECT content FROM templates WHERE id='foo'", 'DB');
```

// or use this syntax:

```
$tmpl->readTemplatesFromInput('templates[@id=foo]/@content', 'DB');
```

It is now also possible to use any PHP function or method as a default value for a variable which allows you to prefill a value with the current timestamp. Furthermore I have been putting a lot of work into the new patTemplate manual.

patForms 0.9.0b3 mostly contains bugfixes and provides several install-groups for the PEAR installer, which allow you to choose your form renderer when installing patForms. This will hopefully be one of the last beta version for patForms 0.9.0 as we plan to release a stable version during the next 4-6 weeks.

Both packages can easily be installed using of PEAR channel server at pear.php-tools.net.

Posted by schst in PHP, PAT at 15:25

Creating PEAR-installable nightly builds

If you incorporate a "release early, release often" policy for your projects, it helps you to detect bugs in an early development stage of your application. In many cases making a release takes some time and you do not release new versions as often as you intended. We had these problems with our projects over at www.php-tools.net. That's why we created snaps.php-tools.net, a site where you can download nightly builds of our projects as ZIP, TAR.GZ or TAR.BZ2 archive. This saves the users, who want to test the latest development versions the hassle checking out the latest version from our subversion repository. The problem of this technique was, that most users use the PEAR installer to deploy our packages and if they downloaded a ZIP file, it's not possible to use the PEAR installer on this file. That's why we came up with a solution to create nightly builds that are installable with the PEAR installer. Read on, if you are interested in how this can be achieved...

To create a PEAR installable package, all you need is a package.xml file, that contains all information about the project and the related files. So to create a nightly build, you will need a current package.xml that describes the package. Creating this by hand can be very cumbersome, as it has to contain all dependencies as well as all files that you want to include in the resulting package. But help has arrived, as Greg Beaver provides a PEAR package that aids you in the creation of package.xml files: PEAR_PackageFileManager. This package provides a PHP-API to read, modify and write the package.xml file needed to create an installable package. To automate this process, you need to follow these steps: Export the latest version from Subversion/CVSDetermine a version number to identify the nightly buildGenerate a package.xml fileUse the pear package command to create the archiveThe first task at hand is extremely easy, as Subversion and CVS both offer the export command. To get the latest version of patTemplate, the following command does the trick:

```
$ svn export http://www.php-tools.net/svn/patTemplate/trunk patTemplate
```

Now that you have exported the latest code-base the next steps is to create a unique version number. We decided that we wanted to use the latest officially released version (e.g. 3.1.0) and add a version suffix. This suffix could either be the latest SVN revision, or the date that the last change has been made.

As the revision in Subversion is incremented for the complete repository, once a commit has been made and all our projects share a common repository, we decided to use the date of the last change in the project folder as the version suffix. Once you have checked out the contents, this can be easily accomplished using the SPL's

```
RecursiveDirectoryIterator:function getLastModifiedDate($folder) {  
    $modified = 0;  
    $dir = new RecursiveDirectoryIterator($folder, RecursiveIteratorIterator::LEAVES_ONLY);  
  
    foreach (new RecursiveIteratorIterator($dir) as $file) {  
        if ($file->getMTime() > $modified) {
```

```
        $modified = $file->getMTime();
    }
}
return $modified;
}
```

This information now has to be used in a script, that will generate the package.xml file. For snaps.php-tools.net, we decided to split these tasks. The SVN export and the generation of the version suffix is handled by a simple script `createSnaps.php` which also build the plain ZIP or TAR archives.

After exporting the latest SVN version, this script searches for a file called `autopackage2.php`. If the file is available, it will execute the file and pass the version suffix as the sole parameter. `$createPackageFile = "php autopackage2.php " . date('YmdHi', $mtime);`
`exec($createPackageFile);`The `autopackage2.php` file then will use `PEAR_PackageFileManager` to create the current package.xml file. I will not go into detail here, you may take a look at an example in the `patTemplate` subversion repository. The most important part is, how the version number is created:// Base version
`$baseVersion = '3.1.0';`

```
// current version
$version = $baseVersion . 'dev' . $argv[1];
```

Furthermore it uses a `filelistgenerator` provided by `PEAR_PackageFileManager` to include all files that have been exported from SVN. This `autopackage2.php` script will generate a `package.xml` file and save it on disk. To create a PEAR installable package from this newly generated `package.xml` file, all that's left to do is execute the following command:`$ pear package -n`The `-n` switch is used to echo the full path to the generated `.tgz` package file after it has been generated. We use it to move this file to the document root, so it can be downloaded. And voilà you created a PEAR-installable nightly build.

For snaps.php-tools.net we also implemented some more features:It is possible to not only create a version from the current trunk, but as many branches as you likeWe also create ZIP, TAR.GZ and TAR.BZ2 archives for users who do not use the PEAR installerWe implemented a simple class that acts as a webfrontend to download the nightly builds.If you are interested in using the same technique for your nightly builds, feel free to use the code of snaps.php-tools.net as a starting point. You can get it from our subversion repository:`$ svn co http://www.php-tools.net/svn/patSnaps/trunk`Beware that some of the code is several years old and while it works with PHP 5.1 it surely could be optimized. Examples for the `autopackage2.php` file are available in the subversion repositories for `patTemplate`, `patForms` and `patBBCode`.

Posted by schst in PHP, PAT at 11:23

Monday, January 3. 2005

HTTP_SessionServer 0.4.0 available

I just released a new version of HTTP_SessionServer, a daemon that is able to store key value pairs and communicates with any kind of client using a simple protocol.

Besides some cosmetic changes, this release features a new backend contributed by Carsten Lucke, which stores session data in a database using the DB abstraction layer.

If you'd like to learn more about how HTTP_SessionServer works, you can take a look at the presentation I did at the International PHP Conference in Frankfurt.

Posted by schst in PAT at 19:47

Wednesday, March 3, 2004

CVS repository gets alive

Currently the pat-team works hard to refill the CVS repository. After the whole project moved to the new server, we got "root"-permissions for the entire system. Now there are many giga bytes on the brand new harddisk to be filled with PHP-stuff.

Besides the new subdomain, you can find some new packages in our CVS-repository at cvs.php-tools.net.

Posted by gERD in PAT at 01:37

Blog Export: a programmer's best friend, <http://blog.php-tools.net/>

Tuesday, February 24. 2004

To bug or not to bug...

Today I installed BugZilla on our server. We will use this tool to track all bugs for all of our projects. The current installation may frighten users that are not familiar with BugZilla, we will be working on the user interface during the next days/weeks.

If you want to report a bug for our tools, just visit bugs.php-tools.net.

Posted by schst in PAT at 22:21

Blog Export: a programmer's best friend, <http://blog.php-tools.net/>

Sunday, February 22. 2004

Whuff! PAT started blogging

Our friendly little dog finally takes part in the new hype and starts his own blog. With the help of his favourite programmers argh, gERD and schst, he will inform you what's going on in his life and the PHP Application Tools Team.

BTW: This blog is powered by the great s9y blog.

Posted by schst in PAT at 19:46