

Friday, January 28. 2005

### **Tutorial for Services\_Ebay**

Adam Trachtenberg, manager of Technical Evangelism at eBay, has written an introductory tutorial on Services\_Ebay, one of my newest PEAR packages.

Although this ONLamp.com tutorial is rather short, it deals with some interesting aspects of Services\_Ebay, as he shows what can be accomplished by object-overloading and he even makes use of the ability to switch model classes in a way I've never thought of them. Adam implements his own models to use them as a presentation layer as he only changes the output of the `__toString()` method.

If you are interested in the Services\_Ebay package, this is the tutorial that will get you started.

Posted by schst in PHP at 15:36

Monday, January 24, 2005

### **International PHP 2005 Conference - Spring Edition**

I will be giving two talks at the annual PHP Conference in Amsterdam this year: Go OO! - Real-life patterns in PHP5  
This talk will show you how to use PHP5's new object-oriented capabilities and the interfaces of SPLComponent and event-driven architectures in PHP5  
This session will introduce you to new concepts of web development and show you how to implement event handling in your applications which make them a lot more flexible  
In addition to these sessions I'll be doing a power workshop entitled "XML and Web services with PHP5 and PEAR" together with Tobias Schlitt.

Hope to see you there!

Posted by schst in PHP at 17:23

Sunday, January 16, 2005

### **PEAR is getting event support**

Bertrand Mansion started a new proposal for Event\_Dispatcher, a package that provides easy-to-use event handling for your projects. I only realized that this package has been proposed as Bertrand started the call for votes on this package, but as I'm using this technique a lot in patPortal and at my work, I decided to give this package a try.

The package's architecture is quite simple: There is an Event\_Dispatcher, which is able to hold observers (native PHP callbacks) for different notification types. These observers will be notified whenever an event (aka notification) occurs. This notification is a simple object that stores information about the event itself. In real life you can use it to add functionality to your sites, without having to change the code of your application, e.g. in my applications, I send a notification, whenever a user places an order (onOrder) and pass the order object to the notification. This way I can easily plug in new objects, that write logfiles, update cookies or send a trackback to an affiliate partner.

As I liked the simple architecture of the package, I played with it a little and reached some limits of the package as it has been missing features I heavily use in my commercial projects. The first feature I was missing was the ability to let an observer cancel the notification (for example if you want to plugin in some checks before a user is able to place an order, which enable you to cancel an order after the user pressed the submit button), so I added this to Bertrand's code. Next was the ability for event bubbling, which allows you unlimited levels of event handling and an event that was triggered at a lower level will then inform all observers of the higher levels. So I added this feature as well, by allowing the user to nest as many dispatchers as they like.

And last but not least, I wanted to be able to use a custom class for notifications, which is now possible in my version as well (for a single dispatcher as well as all dispatchers).

I hope that Bertrand will include all these changes in the official distribution and that this package will be accepted by the PEAR community, so if you haven't voted, do it now.

Posted by schst in PHP at 14:13

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