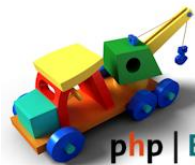


php | Motores de Workflow

Más allá de las Aplicaciones CRUD

Carlos Buenosvinos (carlos.buenosvinos@gmail.com)
Zend PHP Certified – CIO at Latam Training



Workshops
php | Barcelona 2008
07.06.2008

[Grupo phpbarcelona.org](http://Grupo.phpbarcelona.org)

7 de junio de 2008

Carlos Buenosvinos (carlos.buenosvinos@gmail.com)

Introducción

Este documento contiene el taller práctico que se va a desarrollar en el Workshop “Motores de Workflow en PHP (Más allá de las Aplicaciones CRUD)”, dentro del ciclo de Talleres prácticos que organiza el grupo phpbarcelona.org en la phpBarcelona Workshops 2008, el día 7 de Junio.

Objetivos

El Taller pretende mostrar los conceptos básicos de un Motor de Workflow y hacer que el lector los ponga en práctica, a través de ejercicios progresivos. Aunque existen algunas soluciones de motores de Workflow en PHP (Galaxia, CuteFlow, OpenFlow, etc.) todo y que no se ajustan bastante a una API genérica, vamos a utilizar unos componentes del framework eZComponentes de (aZ Sytems SA).

Metodología

El lector deberá ir realizando ejercicio a ejercicio, de manera secuencial. El tiempo aproximado para la realización de todos los ejercicios es de 1 hora.

Requerimientos

Para poder desarrollar los ejercicios de este taller es necesario disponer de:

1. Servidor Web (Apache) con soporte para PHP
2. PHP
3. Servidor MySQL
4. Cliente MySQL

Recomiendo algún entorno de unificado tipo LAMP, XAMP, etc.

Ejercicio 1: Descargar, Instalar y Configurar los ezWorkflow Components (5-10 minutos)

Tanto los componentes como su documentación están disponibles en Internet (<http://ezcomponents.org>) – Podéis seleccionar el link “Click to Download” que descargará el “.bz” o ir a la Sección Downloads para descargar el “.zip”

1. Descargamos los ezComponents de la Web
2. La configuración a la hora de crear una nueva aplicación es personal de cada uno podéis optar por:
 - a. Crear un Host Virtual
 - b. Un simple directorio dentro de vuestro “docroot”, etc.
3. Las componentes son una librería externa, por lo tanto, la podéis incluir en un subdirectorio “lib” dentro de vuestra aplicación o de vuestra carpeta de librerías de terceros
4. Para incluir las componentes, tenemos que configurar el php.ini, directiva para que apunte a la ruta donde se encuentran los eZComponents.

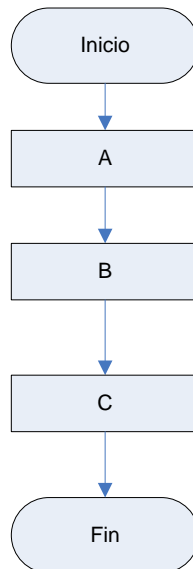
Ejercicio 2: Crear un sencillo Action Node (1-5 minutos)

Dentro de los tipos de Nodos que existen en los eZComponents, los Action Nodes son los que realmente hacen las acciones o actividades que gestiona el propio Workflow. Un Action Node está asociado con un Servicio y eso es lo que vamos a crear.

1. Descargamos el archivo “ServicioImprimir.php.todo”
2. Llenar los agujeros de código para implementar un Servicio llamado “ServicioImprimir” que reciba un mensaje en la constructora y al ejecutarse imprimir ese mismo mensaje.

Ejercicio 3: Programar un Workflow sencillo (5-10 minutos)

En este ejercicio, vamos a crear un Workflow secuencial de 3 pasos utilizando el ServicioImprimir. El diagrama de workflow es el siguiente:



1. Descargar el archivo "ejercicio3.php.todo"
2. Llenar los agujeros de código para implementar programando el Workflow de la figura.
3. Lanzar la ejecución del Workflow definido
4. El resultado debería ser los 3 mensajes secuenciales de los nodos A, B y C respectivamente.

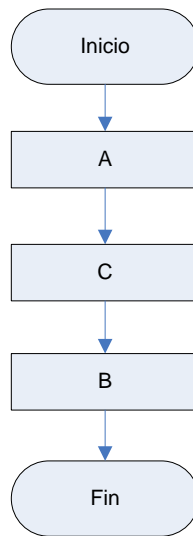
Ejercicio 4: Volcar el Workflow en un fichero XML (1-5 minutos)

En este ejercicio, vamos a generar el archivo XML respectivo al Workflow del Ejercicio anterior:

1. Descargar el archivo “ejercicio4.php.todo”
2. Llenar los agujeros de código para generar el Workflow definido en XML.
3. Abrir el Archivo XML e identificar cada elemento con su Nodo

Ejercicio 5: Modificar el XML y ejecutar el Workflow de Nuevo (5-10 minutos)

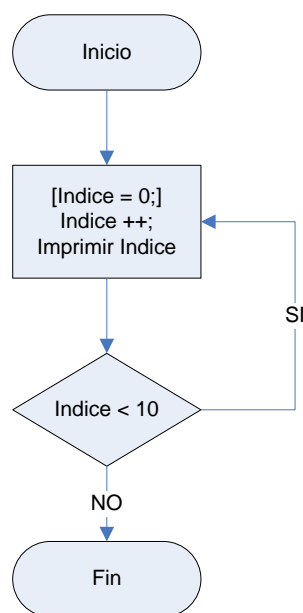
1. Editar el Archivo XML generado en el ejercicio anterior para alterar el orden de los Pasos B y C.
2. Descargar el archivo “ejercicio5.php.todo”
3. Llenar los agujeros de código para generar ejecutar el Workflow definido por el XML modificado. Comprobar que ahora los pasos B y C están intercambiados respecto al primer ejercicio.



Ejercicio 6: Modificando el Estado de una Ejecución a través de las Variables de Estado (10-15 minutos)

Cada ejecución o thread de un Workflow determinado tiene un contexto donde se puede guardar información que los Nodos pueden consultar y activarse en función del estado de dicha información. Para acceder a esta información podemos utilizar los métodos `getVariable`, `setVariable`, `unsetVariable` y `hasVariable` del objeto `ezcWorkflowExecution` que recibe el método `execute` de todos los Servicios.

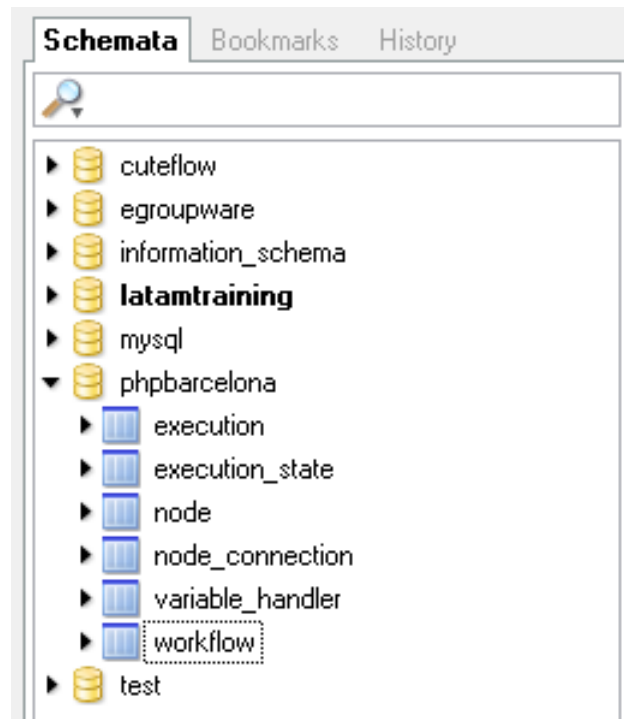
1. Descargar el archivo “ejercicio6.php.todo”
2. Descargar el archivo “ServicioNumerico.php.todo”
3. Llenar los agujeros de código para generar el Workflow de la figura. El resultado tiene que ser un listado del 1 al 10.



Ejercicio 7: Configurar la BBDD para hacer Persistentes los Workflows (1-5 minutos)

Para poder utilizar resumir ejecuciones de Workflows, utilizar subWorkflows, así como Nodos de tipo Input, necesitamos acceso a una BBDD. En este ejercicio vamos a configurar la BBDD (Crear las tablas)

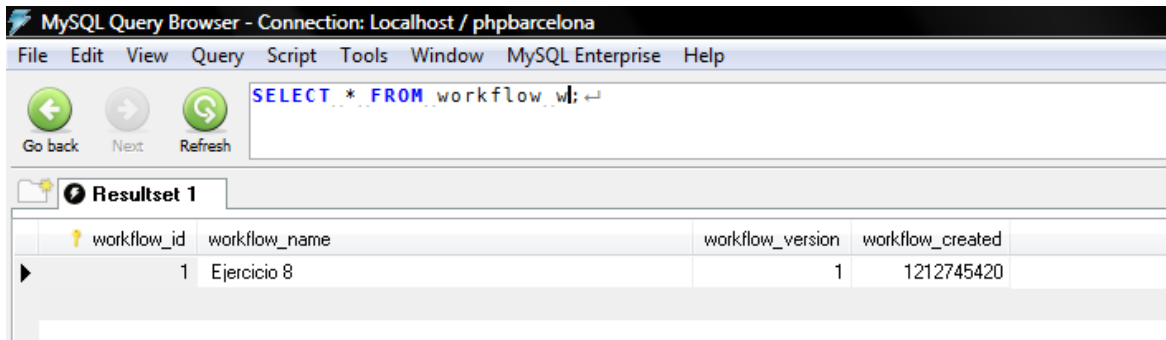
1. Descargamos el archivo “workflow.sql”
2. Ejecutamos las sentencias SQL en nuestro Cliente de MySQL (phpmyadmin, MySQL Query Browser, etc.)



Ejercicio 8: Guardar la definición de un Workflow en la BBDD (5-10 minutos)

Para poder utilizar resumir ejecuciones de Workflows, utilizar subWorkflows, así como Nodos de tipo Input, necesitamos acceso a una BBDD. En este ejercicio vamos a configurar la BBDD (Crear las tablas)

1. Descargar el archivo “ejercicio8.php.todo”
2. Dibujar a mano la Representación gráfica del Workflow
3. Llenar los agujeros de código para guardar el Workflow en la BBDD
4. Comprobar que la tabla “workflow” dispone de nuestro workflow insertado



The screenshot shows the MySQL Query Browser interface. The title bar reads "MySQL Query Browser - Connection: Localhost / phpbarcelona". The menu bar includes "File", "Edit", "View", "Query", "Script", "Tools", "Window", "MySQL Enterprise", and "Help". Below the menu bar are three navigation buttons: "Go back", "Next", and "Refresh". The query editor contains the SQL statement: `SELECT * FROM workflow_wl;`. Below the query editor, a "Resultset 1" tab is active, displaying a table with the following data:

workflow_id	workflow_name	workflow_version	workflow_created
1	Ejercicio 8	1	1212745420

Ejercicio 9: Cargar la definición del Workflow anterior de la BBDD (5-10 minutos)

En las Aplicaciones Web basadas en Motores de Workflow, lo normal es obtener una definición de la BBDD, y crear o resumir una ejecución de dicho Workflow. En este ejercicio, vamos a cargar la definición insertada en el Ejercicio anterior y vamos a lanzar una ejecución

1. Descargar el archivo “ejercicio9.php.todo”
2. Llenar los agujeros de código para cargar la definición del Workflow anterior de la BBDD
3. Crear un entorno de ejecución y lanzar un nuevo thread del Workflow. (el método start devuelve un parámetro que representa el ID del thread, debéis imprimirlo por pantalla para su uso posterior en el siguiente ejercicio)
4. ¿Por qué no aparece nada? (Aparte de la representación del Workflow)
5. Después de lanzar la ejecución, imprime por pantalla el estado de la Ejecución (métodos isResumed, isSuspended, etc.)
6. ¿Cómo reacciona una ejecución cuando llega a un Nodo tipo Input?
7. Haz un print_r del resultado de la invocación al método getWaitingFor() sobre el objeto de la Ejecución
8. ¿Qué parámetros está esperando?

Ejercicio 10: Continuar con la Ejecución anterior con los parámetros que necesita (1-5 minutos)

Resulta que en el ejercicio anterior, la ejecución se quedó parada esperando cierta variable de estado para poder continuar. En este ejercicio, vamos a cargar esa ejecución pendiente (necesitaremos su ID) y vamos a resumirla dándole los parámetros que necesita.

1. Descargar el archivo “ejercicio10.php.todo”
2. Llenar los agujeros de código para cargar la definición del Workflow anterior de la BBDD
3. Crear un entorno de ejecución y continuar (`resume()`) el thread del Workflow con la variable que faltaba y vimos en el ejercicio anterior.

Ejercicio 11: Práctica

Adquiridos algunos conceptos hasta aquí, el objetivo de este ejercicio abierto, es desarrollar una pequeña aplicación que gestione el proceso de Validación de un documento. Os dejo el enunciado en Inglés. Como la selectividad podéis elegir cualquiera de las dos opciones.

1. Podéis utilizar lo que queráis
2. Cuando lo acabéis (tanto en el Workshop como fuera de él) me lo podéis enviar a carlos.buenosvinos@gmail.com
3. Tendremos regalitos para aquellos que lo envíen y la solución funcione.

Multiple Approval, ISO Certification

^^

This scenario is from a current customer of the eZ Publish ECMS providing quality assurance for dairy products. The customer has information about the dairy products stored in eZ Publish. When they update any content there is a strict ISO-governed process to follow. This process basically consists of a five-level approval:

- Bertrand produces an article.
- Approver Level 1: Bård decides who the next four approvers are.
He can also edit the article and send it back to its creator.
- Approver Level 2: Melissa reviews the article for political correctness.
She can edit the article and send it back one level.
- Approver Level 3: Vidar reviews the article for sales arguments.
He can edit the article and and send it back one level.
- Approver Level 4: Jennifer does grammar checks on the article.
She can edit the article and and send it back one level.
- Publisher: Markus approves the final article and chooses the time and location for publication.

It is possible to see all on-going processes for an administrator. He or she can see each article as well as its state and which person currently

handles it.

Employment Process

^^^^^^^^^^^^^^^^^^^^^^^^^^

This scenario is from the intranet of a current customer of the eZ Publish ECMS and is used when a new employee is hired.

- One person creates an Employee object (including name, address, email, etc.).
- An e-mail with a link for final approval of the employment is sent to the CEO.
- Once the CEO has approved the new employment three parallel activities are started:
 - An e-mail to the system administrator is sent with the request to create e-mail and other accounts.
The e-mail contains a link for the system administrator to click when he is done.
 - An automatic process is started to set up accounts on different systems.
 - An e-mail to the administration is sent with the request to buy new hardware for the new employee.
- Once these three activities have been completed, the workflow continues.
- The Employee object is published.
- An e-mail with detailed information is sent to the new employee.

The on-going status for all employment processes at any time is available to anyone with the appropriate permissions.