

Wednesday, 19 January 2005

Gestion d'évènements dans PEAR

Traduction du billet 'PEAR is getting event support' de php-tools:

Bertrand Mansion (en) a proposé une nouvelle librairie pour inclusion dans PEAR, `Event_Dispatcher` (en). Celle-ci offre une gestion simple d'évènements facile à intégrer dans une application PHP. Bertrand a déjà démarré l'appel au vote (en), mais comme nous utilisons déjà cette technique dans `patPortal` (en), et Stéphane dans son boulot chez 1et1 allemagne (de), il l'a essayé:

L'architecture de la librairie est assez simple: la classe `Event_Dispatcher` peut contenir des observateurs (appels natifs PHP) pour divers types de notifications. Ces observateurs sont informés de chaque évènement par une notification sous la forme d'un objet (la notification) très simple qui contient des informations sur l'évènement. Dans la vie de tous les jours, cette technique permet d'ajouter des fonctionnalités à une application existante sans avoir à modifier le code source. Dans le site de 1et1 dont s'occupe Stéphane, il envoie une notification à chaque commande (évènement `onOrder`) en y ajoutant l'objet gérant la commande comme information. De cette manière, il est facile d'intégrer nombre d'objets pour par ex. écrire des logs, mettre à jour des cookies ou d'envoyer des suivis à des partenaires affiliés.

L'architecture simple de `Event_Dispatcher` est un atout, mais on arrive vite à des limites - dans ses essais, Stéphane a trouvé quelques lacunes. La première chose est qu'un observateur ne peut pas annuler (arrêter) un évènement: cela permettrait par ex. d'ajouter une ultime vérification avant que la commande d'un client ne soit envoyée, et d'annuler la commande le cas échéant. La deuxième chose est que la remontée d'évènements ('event bubbling' en anglais) n'est pas possible: cela permettrait des niveaux illimités de gestion d'évènements, chaque évènement pouvant remonter toute la hiérarchie des observateurs. Enfin, il n'est pas possible de définir une classe sur mesure pour les notifications (ce qui peut être nécessaire selon le type d'informations qu'on veut y stocker).

Stéphane a implémenté toutes ces fonctionnalités dans le code de Bertrand, en espérant qu'il voudra bien inclure ces changements dans la version officielle de `Event_Dispatcher` lors de sa publication (si elle est acceptée par la communauté des développeurs de PEAR... votez!).

Posté par Argh dans php-tools à 13:21