

easyObject

modern web applications made easy

Dossier de présentation

version 1.0 - décembre 2012

par Cédric François

<http://www.cedricfrancoys.be/easyobject>

Ce document est mis à disposition selon le Contrat Attribution-NonCommercial-ShareAlike 3.0 Unported
disponible en ligne <http://creativecommons.org/licenses/by-nc-sa/3.0/>
ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Table des matières

En bref.....	3
Aspects techniques.....	3
En détails.....	4
Introduction.....	4
Principes de base.....	4
Pourquoi PHP ?.....	4
Pourquoi un nouveau projet ?.....	4
Philosophie.....	6
Origine du projet.....	6

En bref

easyObject est une solution dont le cœur applicatif, écrit en PHP5, est un ORM (Object-Relational Mapping) qui permet d'assurer la persistance et la manipulation d'objets en associant des classes aux tables d'une base de données relationnelle. La particularité d'easyObject est de proposer une API indépendante du langage utilisé, ce qui en fait une solution particulièrement adaptée aux applications web. C'est en quelque sorte un « ORM orienté-web » ou WORM.

easyObject est une solution qui :

- se veut accessible au plus grand nombre ;
- ne nécessite pas d'investir beaucoup de temps dans son installation et dans son apprentissage ;
- permet d'assurer très facilement les tâches communes à la plupart des applications web (*support de tout type de contenu, définition de permissions, gestion des brouillons, suivi des versions, mise en cache des contenus, support multilingues pour les contenus et l'interface utilisateur, réécriture des URL, journalisation, suivi du statut de publication, récupération des suppressions accidentelles*).

easyObject est mis à disposition sous licence open source GNU GPL 3.0.

Aspects techniques

Configuration logicielle très répandue

- PHP 5.3 +, configuration par défaut avec support MySQL
- Apache 1.3 +, configuration par défaut avec support PHP
- MySQL 5.1+

Installation triviale

- 1) télécharger
- 2) décompresser
- 3) personnaliser le fichier de configuration

Mise en œuvre très intuitive

3 types d'opérations

- requêtes DO : faire quelque chose (créer, modifier, supprimer, ...)
- requêtes GET : renvoyer quelque chose (une liste, un champ, une traduction, ...)
- requêtes SHOW : afficher quelque chose (une page, un formulaire, un graphique, une application, ...)

4 fonctions de base

- browse : obtenir la valeur des champs d'un ou plusieurs objets
- search : trouver les objets répondant au filtre de recherche
- update : modifier la valeur des champs d'un ou plusieurs objets / créer un nouvel objet
- remove : supprimer un ou plusieurs objets

3 langages impliqués pour le développeur

- PHP : être familiarisé avec la syntaxe de base est suffisant (pour le reste, il y a les exemples et la doc)
- HTML : ceux qui savent créer des balises DIV et SPAN s'en sortiront
- JS/JSON : la connaissance de Javascript est facultative et JSON est plus une syntaxe qu'un langage

En détails

Introduction

easyObject est un outil à la fois simple et robuste, conçu pour une prise en main rapide, et dont le but est de permettre d'assurer facilement les tâches communes à la plupart des applications web modernes. Concrètement, il s'agit d'un framework open source pour applications web, s'apparentant à un ORM, écrit en PHP5 et utilisant jQuery pour l'interface utilisateur, et inspiré par les initiatives open source OpenERP, easyCMS et KnINE.

Principes de base

Le rôle principal de la plateforme easyObject est d'assurer la persistance des objets en associant des classes aux tables d'une base de données relationnelle. C'est-à-dire d'établir une correspondance entre des objets (les instances de classe) et des enregistrements (les lignes d'une table).

L'objectif étant de permettre au développeur d'application de se concentrer sur la définition des objets et de leurs interactions plutôt que sur la couche logicielle permettant leur persistance et leur gestion.

Ainsi, cet outil met à disposition du développeur web une série de méthodes utilisables avec n'importe quel langage (pour l'instant PHP et Javascript) permettant des manipulations d'objet d'une grande simplicité.

Pourquoi PHP ?

PHP étant un langage orienté web largement répandu et les plateformes LAMP (Linux, Apache, MySQL, PHP) étant très courante, l'installation est, dès le départ, grandement simplifiée. Par ailleurs, les fonctionnalités orienté-objet désormais offertes par PHP (5.3+) permettent un niveau satisfaisant de souplesse et de rigueur. Enfin, la partie serveur est écrite en PHP mais l'interface phpRPC permet d'utiliser l'API avec n'importe quel langage (c'est d'ailleurs le cas par défaut pour l'interface graphique qui fait appel à l'API easyObject au travers des fonctionnalités ajax de javascript).

Pourquoi un nouveau projet ?

Bonne question, vu qu'il existe déjà des projets open source écrits en PHP implémentant un mécanisme de persistance des objets (CakePHP, Doctrine, FuelPHP, pdoMap, Propel, ...).

Cette initiative se distingue par le fait qu'elle est issue de la volonté de proposer une solution :

- qui soit accessible, fiable, légère et robuste, à la fois dans sa conception et dans son utilisation ;
- qui couvre de manière intrinsèque les besoins communs à la majorité des applications web (*support de tout type de contenu, définition de permissions, gestion des brouillons, suivi des versions, mise en cache des contenus, support multilingues pour les contenus et l'interface utilisateur, réécriture des URL, journalisation, suivi du statut de publication, récupération des suppressions accidentelles*) ;
- qui prenne en charge tous les aspects liés à la gestion des objets (définition, interactions, consultation, édition, validation) ;
- qui ne dépende pas d'une entreprise commerciale.



Philosophie

Par ailleurs, l'idée maîtresse de ce projet est la simplicité. Dans les grandes lignes, cela donne :

- **Niveau d'abstraction élevé**
L'idée est de garder le code source aussi léger que possible. Pour compenser la complexité du code et faciliter sa compréhension, celui-ci est rigoureusement commenté et documenté.
- **Nombre de langages minimal**
Le nombre de langages impliqués est gardé le plus bas possible (PHP, SQL, Javascript, JSON, HTML) et, en particulier, le nombre de langages que le développeur-utilisateur doit manipuler est limité à deux : PHP et HTML.
- **Grande flexibilité**
En dehors de quelques contraintes minimales, le développeur-utilisateur est libre de faire les choses de la manière qu'il juge la plus appropriée ou en fonction de ses besoins applicatifs particuliers (voire même dans le langage qu'il choisit d'utiliser).
- **« Fail-safe » mais PAS « foolproof »**
La responsabilité d'une utilisation correcte du code est laissée au développeur-utilisateur.
 - o L'application n'est pas conçue pour fonctionner coûte que coûte : si quelque chose tourne mal, l'application ne plante pas mais le script s'interrompt en indiquant (gentiment) ce qui s'est passé.
 - o Pas d'automatisation : par exemple, aucune synchronisation ou vérification de correspondance n'est faite entre les définitions des classes PHP et en base de données.

Note : des scripts stand-alone (dont les actions ne sont pas couvertes par l'API de base) permettent, entre autre, de réaliser les tâches suivantes :

- Créer une base de données sur base d'un schéma SQL
- Générer une classe PHP à partir d'une table existante
- Vérifier la cohérence entre DB et classes
- Vérifier la syntaxe des classes, des templates et des fichiers de traduction

Origine du projet

Bien que easyObject soit une solution dont le cœur applicatif est un ORM, la vocation première de la démarche à l'origine du projet est le développement d'outils collaboratifs orientés web se déclinant autour des thèmes de l'entraide, de la réappropriation des savoirs et de l'évolution responsable.

Ces outils n'étant pas clairement définis et devant pouvoir évoluer rapidement, l'idée est venue de développer une solution qui puisse répondre aux besoins d'applications qui pourraient être évolutives, qui ne seraient pas forcément très complexes, qui n'auraient pas nécessairement une durée de vie importante, qui devraient pouvoir être développées rapidement et maintenues par des volontaires sans grande expérience dans le développement d'applications.

Il fallait une solution présentant une grande flexibilité et permettant d'assurer facilement les fonctions de base d'une application web.

Étant donné que le résultat est effectivement assez intuitif, plutôt robuste et très polyvalent, il semblait tout naturel de le rendre accessible au plus grand nombre.

Le projet était open source dès le départ, mais des efforts supplémentaires ont été faits pour en faire une véritable solution libre (code disponible sur des plateformes de diffusion, documentation, systèmes de validation).

