SQLi Import

Présentation

21 / 09 / 2010

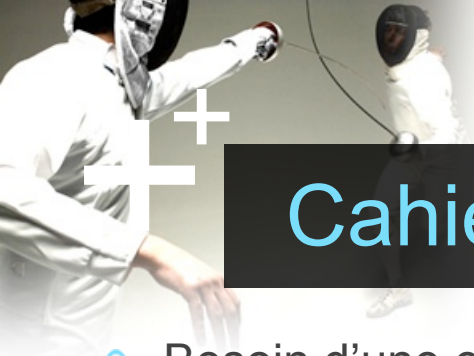
Sommaire

- Constatations
- Cahier des charges
- Principales fonctionnalités
- API simplifiée
- Démonstration

Constatations
&
Cahier des charges

Constatations

- L'import de contenu est une question récurrente
  - Imports «one-shot» (transferts de données)
  - Contenus partenaires (fournisseurs de contenu, flux d'affiliation...)
  - Harmonisation du SI (catalogue produit, interfaçage ERP)
- Formats divers
  - XML
  - CSV
  - SOAP
- API de gestion de contenu puissante mais très complexe
  - Gestion des langues
  - Versions, sections, états d'objet
  - Types de données variés
  - Gestion du cache
- Besoin de performances et de flexibilité
- Une extension déjà existante : data_import
  - De très bonnes idées, mais peu flexible
  - Code vieillissant



Cahier des charges

- Besoin d'une extension robuste
  - Capacité de gérer de très grandes quantités de données
  - Bonne gestion d'erreur afin de ne pas bloquer une chaîne d'imports
- Besoin d'une extension flexible
  - Doit pouvoir s'adapter à tout type de données
  - Facile à configurer
- Besoin d'une extension facilitant l'exploitation
  - Suivi des imports en cours
  - Programmation d'imports récurrents
- Besoin d'une extension facile à implémenter
  - Documentation
  - API simplifiée (éviter les mauvaises pratiques)
  - Interface simple

SQLi Import :
Principales fonctionnalités

SQLi Import : Principales fonctionnalités

- Gestion de «handlers» d'import
  - Chaque source de données est gérée par un handler écrit en PHP
  - Interface PHP permettant d'encadrer le handler
  - Gestion de différents formats (parseurs CSV et XML fournis)
  - Itération sur tous les éléments de la source à importer
- API simplifiée
- Configuration simple et options à la volée
  - Chaque handler possède une section de configuration dans *sqliimport.ini*
  - Possibilité de passer des options à la volée (hors configuration)
- Gestion d'erreurs
  - Si une exception est attrapée lors d'une itération, une erreur est loguée et le programme passe à l'itération suivante
- Exploitation
  - Possibilité de lancer des imports «one-shot»
  - Programmation d'imports récurrents
  - Possibilité d'interrompre un import en cours sans risque
  - Suivi des imports en cours (pourcentage de progression, notes de progression)
  - Traçabilité des utilisateurs & gestion de droits
  - Performances (désactivation du cache et activation du DelayedIndexing, de manière temporaire)

SQLi Import :
API Simplifiée

SQLi Import : API Simplifiée

- L'API d'eZ Publish est très puissante mais aussi très complexe
  - Mal connue et souvent mal maîtrisée par les développeurs
  - Difficulté d'apprentissage
  - Mauvaises pratiques
- API simplifiée prévue dans les futures versions d'eZ Publish
  - Matterhorn (4.5) ?
  - Les besoins sont réels, concrets et immédiats
- Prenons les devants !
  - API de SQLi Import compatible avec les concepts déjà présentés en juin 2010
  - Compatible eZ Publish 4.1+
  - Totalement objet, se basant sur l'approche des Apache Zeta Components

```php
$cli->notice( 'Creation of a new "comment" object' );
$options = new SQLIContentOptions( array(
    'class_identifier'      => 'comment',
    'remote_id'             => 'my_ubber_cool_remote_id',
    'language'              => 'fre-FR'
) );
$comment = SQLIContent::create( $options );
$cli->notice( 'Current version : '.$comment->current_version );
$comment->fields->subject = 'Mon super sujet';
$comment->fields->author = 'Moi !';
$comment->fields->message = 'Le commentaire de la mort';

$comment->addTranslation( 'eng-US' );
$comment->fields['eng-US']->subject = 'My great subject';
$comment->fields['eng-US']->author = 'Batman';
$comment->fields['eng-US']->message = 'Death comment';

$comment->addLocation( SQLILocation::fromNodeID( 2 ) );
$comment->addLocation( SQLILocation::fromNodeID( 43 ) );

$publisher = SQLIContentPublisher::getInstance();
$publisher->publish( $comment );

$cli->notice( 'Current version : '.$comment->current_version );

// Loop against locations
foreach( $comment->locations as $nodeID => $location )
{
    $cli->notice( $nodeID.' => '.$location->path_string );
}
```

SQLi Import : Démonstration

Questions ?

- Page du projet : http://projects.ez.no/sqliimport
- Mon Blog : http://www.lolart.net
- Ma Société : http://www.sqli.com
- Twitter : @jvieilledent
- Skype : lolautruche
- Email / GTalk : lolautruche@gmail.com