

0.1 Module : Perl

Dans ce document, je vais détailler l'installation et l'utilisation de perl en tant que langage de script côté serveur. Pour cet usage, aussi appelé CGI (Common Gateway Interface), vous pouvez, soit utiliser le module cgi, natif d'Apache comme indiqué ICI¹ soit utiliser mod_perl, comme indiqué dans ce document. Avec mod_perl, Perl est intégré au serveur Apache. C'est pourquoi il ne réclame pas de nouveaux processus, contrairement à mod_cgi. Autre différence avec mod_cgi : en utilisant mod_perl, vos scripts seront compilés une fois pour toutes (sauf si modification bien-sûr) et resteront en mémoire. Enfin, grâce à mod_perl vos variables globales sont réellement globales. Vous l'aurez compris, les scripts perl sont plus rapides avec mod_perl qu'avec mod_cgi, mais pour utiliser mod_perl, il faudra l'installer vous-même, alors que mod_cgi est livré avec Apache. Pour finir, ce qui suit concerne Apache 1.x, il faudra peut-être quelques adaptations pour le transposer à Apache 2.x

0.1.1 1. Historique et présentation :

Perl est l'abréviation de Pratical Extraction and Report Language (Langage pratique d'extraction et de génération de rapports) ou Pathologically Eclectic Rubbish Lister (Collectionneur Pathologique de Déchets Variés), au choix. Les deux définitions sont acceptées par Larry Hall, le linguiste qui l'a inventé en 1986. Perl est aujourd'hui en version 5.8.0, disponible sous Artistic License ou GNU General Public License, au choix. C'est un langage de programmation de haut niveau, impératif et hautement portable. Il rend les choses faciles encore plus faciles et les choses délicates, possibles. Il est optimisé pour manipuler des caractères dans des fichiers ou non, et conviendra parfaitement aux administrateurs avancés. Il est très extensibles et dispose de nombreux modules. Il a inspiré de nombreux langages, très en vogue comme Python ou PHP.

0.1.2 2. Installer la distribution perl

0.1.3 2.1 Depuis les paquetages de votre distribution

Votre distribution est nécessairement livrée avec un paquetage perl, c'est ce dernier que je vous conseille d'installer, surtout s'il est récent. La dernière version étant la 5.8.0 et datant de juillet 2002. Pour savoir si perl est installé, tapez : perl -v. Dans tous les cas, vous pouvez l'installer vous-même, en suivant ce qui suit.

- Pour les utilisateurs de Debian, tapez :

```
# apt-get install perl
```

- Pour les utilisateurs de Mandriva, tapez :

```
# urpmi perl
```

- Pour les utilisateurs de Redhat, installez l'archive dispo sur votre cdrom :

```
# rpm -Uvh perl-xxxx.rpm
```

- Pour les utilisateurs de Slackware, installez l'archive dispo sur votre cdrom :

```
# installpkg perl-xxxx.tgz
```

¹ <http://www.trustonme.net/didactels/230.html>

0.1.4 2.2 Depuis les sources

Si votre distribution ne fournit pas nativement perl dans une version récente (5.6.x au moins), changez-en !

0.1.5 3. Installer et configurer mod_perl :

0.1.6 3.1 Pré-requis :

Il faut bien-sûr avoir un serveur Apache installé et fonctionnel. Si vous souhaitez, utiliser DBI, il vous faut également un SGBD (MySQL, PostgreSQL ...).

0.1.7 3.2 Installation

Commencez par télécharger la dernière version de mod_perl (1.x) ICI². Pour l'installation :

```
$ tar -xzf /où_est/mod_perl-1.27.tar.gz
$ cd mod_perl-1.27/
$ perl Makefile.PL USE_APXS=1 PREFIX=/usr WITH_APXS=/usr/sbin/apxs EVE-
  RYTHING=1
$ make
# make install
```

0.1.8 3.3 Configurer apache pour perl

Maintenant que mod_perl est installé, il vous reste à modifier la configuration d'Apache, pour que ce dernier le reconnaisse et l'utilise. Je vais supposer dans la suite que votre fichier de configuration d'Apache est /etc/httpd/conf/httpd.conf, que la racine de votre site est /home/httpd/html/ et qu'Apache est lancé, chez vous, par l'utilisateur nobody du groupe nobody. Si ce n'est pas ainsi chez vous, adaptez ! Le répertoire qui contiendra l'ensemble des scripts perl est /home/httpd/perl, créez-le par :

```
#mkdir /home/httpd/perl
#chown -R nobody.nobody /home/httpd/perl
```

Editez maintenant, le fichier /etc/httpd/conf/httpd.conf et assurez-vous d'y avoir ceci :

```
LoadModule perl_module libexec/libperl.so
AddModule mod_perl.c
```

En dehors des balises : <IfDefine SSL></IfDefine> (si vous avez installé mod_ssl). Il faut également que ce fichier contienne ceci :

```
<IfModule mod_perl.c>
Alias /perl/ /home/httpd/perl/
<Location /perl/*.pl>
SetHandler perl-script
PerlHandler Apache : :Registry
Options -Indexes ExecCGI
```

²<http://perl.apache.org/dist/>

```
PerlSendHeader On
</Location>
</IfModule>
```

Avec ça, seuls les fichiers d'extension .pl et contenu dans /home/httpd/perl/ seront identifiés comme des scripts perl. Pour que les modifications soient prises en compte, il suffit de redémarrer apache.

0.1.9 4. Installer les modules additionnels

Sur certaines distributions, les modules listés ici sont déjà installés ou disponibles sur vos cdroms, préférez cette solution pour les installer, si besoin. L'ensemble des modules perl est accessible ICI³. Je suppose dans la suite que la distribution perl est dans /usr/lib/perl5/ (l'exécutable étant /usr/bin/perl), adaptez si nécessaire, à ce qu'il y'a chez vous.

0.1.10 4.1 Installer DBI

Le module DBI, autorise à vos scripts perl à se connecter à la plus part des SGBD du marché. Il faudra, simplement, par la suite, installer le pilote DBD spécifique à votre SGBD. Avant d'installer, le module DBI, vérifiez qu'il n'est pas déjà installé, tapez :

```
find /usr/lib/perl5/ -name DBI.pm
```

Si cette recherche restait sans réponse, c'est que vous n'avez pas DBI installé. Il vous suffit de l'installer comme suite :

```
$ tar -xzvf /où_est/DBI-1.37.tar.gz
$ cd DBI-1.37/
$ perl Makefile.PL
$ make
$ su
Password :
# make install
```

0.1.11 4.2 Installer DBD-mysql

Comme vous vous en doutez, DBD-mysql est le pilote spécifique à MySQL, si vous n'avez pas ou n'utilisez pas MySQL, inutile de l'installer. Avant d'installer DBD-mysql, vérifiez qu'il n'est pas déjà installé, pour cela tapez :

```
find /usr/lib/perl5/ -name mysql.pm
```

Si cette recherche restait sans réponse, c'est que vous n'avez pas DBD-mysql installé. Il vous suffit de l'installer comme suit :

```
$ tar -xzvf /où_est/DBD-mysql-2.1026.tar.gz
$ cd DBD-mysql-2.1026/
$ perl Makefile.PL
$ make
$ su
Password :
# make install
```

³ <http://www.cpan.org/modules/0lmodules.index.html>

0.1.12 4.3 Installer DBD-Pg

Comme vous vous en doutez, DBD-Pg est le pilote spécifique à PostgreSQL, si vous n'avez pas ou n'utilisez pas PostgreSQL, inutile de l'installer. Avant d'installer DBD-Pg, vérifiez qu'il n'est pas déjà installé, pour cela tapez :

```
# find /usr/lib/perl5/ -name Pg.pm
```

Si cette recherche restait sans réponse, c'est que vous n'avez pas DBD-mysql installé. Il vous suffit de l'installer comme suite :

```
$ tar -xzf /où_est/DBD-Pg-1.22.tar.gz
$ cd DBD-Pg-1.22/
$ perl Makefile.PL
$ make
$ su
Password :
# make install
```

0.1.13 4.4 Installer CGI :

C'est une bibliothèque qui vous permet de créer/traiter des formulaires ou générer tout ou partie d'une page html. Il est indispensable pour utiliser perl comme langage de script, côté serveur. Il est généralement installé nativement avec perl, tapez :

```
# find /usr/lib/perl5/ -name CGI.pm
```

Pour vous en assurez. S'il n'était pas installé, téléchargez la version la plus récente ICI⁴. Installez-la par :

```
$ tar -xzf /où_est/CGI.pm.tar.gz
$ cd CGI.pm-2.93/
$ perl Makefile.PL
$ make
$ su
Password :
# make install
```

0.1.14 5. Tests et documentation :

0.1.15 5.1 Tests :

Le but ici n'est ni de faire un cours sur perl ni de présenter sa richesse en fonctionnalités. Je me contente de donner un script dont j'explique le fonctionnement et j'essaie de le rendre aussi lisible que possible, promis. D'abord les explications, ce script présente un formulaire qui vous demande un login, puis il interroge une base de données mysql à la recherche d'informations, qu'il affiche. Pour ce faire, créez un fichier /home/httpd/perl/prog1.pl et rendez-le exécutable :

```
# chmod 755 /home/httpd/perl/prog1.pl
```

Ce fichier contiendra ce qui suit. Commençons par les commandes de configuration :

⁴<http://stein.cshl.org/WWW/software/CGI/CGI.pm.tar.gz>

```
#!/usr/bin/perl -w
use strict;
use DBI;
use CGI;
my $mycgi = new CGI;
```

#!/usr/bin/perl indique au système quel exécutable utilisé, le **-w** c'est pour avoir également les warnings, **use strict** empêche d'utiliser des variables non déclarées, c'est mieux pour un usage en **CGI**. **use DBI** et **use CGI** activent le support respectivement de DBI et CGI, supprimez la ligne inutile, si nécessaire. Passons aux entêtes :

```
print $mycgi->header, $mycgi->start_html(-title=>'Page
  de test',
  -author=>'kernel@trustonme.net',
  -meta=>{'keywords'=>'Home sweet Home'},
  -style=>{'src'=>'http://localhost/css/index.css'});
print '<table width="90%" cellpadding="0" cellspacing="2"
  align="center">
<tbody><tr><td valign="top">'. "\n";
```

ça se passe de commentaire, grâce à mon objet \$mycgi, je génère les entêtes de ma page, plus bas j'affiche le début d'un tableau. Voyons maintenant le coeur du script :

```
if ($mycgi->param()) {
my $login = $mycgi->param("login");
my $dbd = DBI->connect('DBI :mysql :trustonmenet','Kernel','mdp',
  {PrintError => 1, RaiseError => 1}) or die 'connect :',
  $DBI : :errstr;
my $reqdb = $dbd->prepare("SELECT distrib, ville, pays
  FROM inscrit WHERE pseudo = '$login'") or die 'prepare :',
  $dbd->errstr;
$reqdb->execute or die 'execute :', $dbd->errstr;
my $distrib;
my $ville;
my $pays;
($distrib, $ville, $pays) = $reqdb->fetchrow_array;
$reqdb->finish;
$dbd->disconnect;
print $mycgi->p("Bonjour $login, vous utilisez une $distrib,
  vous habitez la ville de $ville ($pays)");
} else {
print $mycgi->start_form();
print 'Entrez votre login : ', $mycgi->textfield('login'),
  '<br />';
print $mycgi->submit('Valider'), $mycgi->reset('Effacer');
print $mycgi->end_form();
}
```

Le if vérifie le formulaire, le else l'affiche. La première ligne de l'if récupère naturellement, les informations entrées dans le formulaire. Pour un affichage personnalisé, je me connecte à un SGBD mysql nommée trustonmenet, mon login est Kernel. Plus bas, je récupère les valeurs dont j'ai besoin et ferme la connexion. Je termine par l'affichage des valeurs obtenues. Terminons par la fin :

```
print '</td></tr></tbody></table>';  
print $mycgi->end_html;
```

Si vous n'avez pas de bdd, effacer les lignes entre la connexion et la deconnexion au SGBD et utilisez plutôt pour l'affichage :

```
print $mycgi->p("Bonjour $login");
```

Pour exécuter ce script, entrez comme adresse dans votre navigateur : <http://localhost/perl/prog1.pl>⁵. N'hésitez pas afficher le code source de la page (**clic-droit -> view source**, depuis votre navigateur), pour lire ce que produisent les fonctions du module CGI. Pour être vraiment complet, notez que vous pouvez appeler ce script directement par : <http://localhost/perl/prog1.pl?login=Kernel>⁶. Dans ce cas, il ne vous proposera pas de formulaire.

0.1.16 5.2 Documentation :

Voilà ! votre serveur Apache devrait reconnaître sans difficulté les scripts perl. Mieux, vous avez désormais la possibilité de manipuler des formulaires html et des bases de données depuis de simple scripts perl. Voici un peu de documentation pour vous faire la main :

- La documentation sur perl, notamment les faqs perl ICI⁷
- La documentation sur l'usage du module CGI est disponible ICI⁸ en Français.
- Avec mod_perl, comme avec mod_cgi, perl honore les variables d'environnement d'apache, via l'encapsulation \$ENV comme indiqué ICI⁹

⁵<http://localhost/perl/prog1.pl>

⁶<http://localhost/perl/prog1.pl?login=Kernel>

⁷http://www.alianwebserver.com/informatique/langages/perl/faq_perl.htm

⁸<http://www.enstimac.fr/Perl/ModulesFr/CGI.html>

⁹<http://www.trustonme.net/didactels/230.html#varapche>