Robots Class
Documentation

version 1.1

Pierre Fauque, <*pierre@fauque.net*>, 0xF8CC4747

28 février 2022

Synopsis

Robots Class est une classe PHP permettant de suivre les visites des robots, webspiders et webcrawlers sur un site web. Les visites peuvent ne rien provoquer du tout, s'enregistrer dans un fichier, une base de données ou être notifiées par mail. C'est ce que fait la version initiale 1.0

Cette version 1.1 prend en compte différents sites. Si vous disposez d'un VPS (Virtual Private Server) sur lequel il y a plusieurs sites web (hôtes virtuels), le suivi peut se faire sur tous les sites hébergés.

—===ooO$Ooo===—

Robots Class is a PHP class for tracking visits by robots, webspiders and webcrawlers to a website. Visits can cause nothing at all, registration into a file, into a database or be notified by email. This is what the initial version 1.0 does.

This version 1.1 takes into account different sites. If you have a VPS (Virtual Private Server) on which there are several websites (virtual hosts), monitoring can be done on all hosted sites.

Table des matières

1 Documentation française

1.1 Le fichier `robots.txt`

S'il est présent, ce fichier texte doit obligatoirement se trouver dans le répertoire racine du site web. S'il est absent ou vide, les robots auront accès à tous les fichiers et répertoires du site pour les indexer. Si vous ne souhaitez pas qu'une page ou un répertoire soit explorés par les robots, il faudra écrire les règles que vous souhaitez dans un fichier `robots.txt`

Les règles comportent la désignation du robot (`User-agent`) et les permissions ou interdictions (`Allow`, `Disallow`). Dans la désignation du robot, l'astérisque signifie « tous les robots » et les permissions ou interdictions commencent à la racine du site.

Ci-dessous trois exemples de fichiers `robots.txt` :

`User-agent: *` `Disallow: /`	Tous les robots seront interdits à partir de la racine
`User-agent: Googlebot` `Disallow: /private/`	Le robot Googlebot ne devra pas explorer le répertoire /private/

`User-agent: YandexBot`
`User-agent: SemrushBot`
`Disallow: /private/`
`Disallow: /members/users.html`
 	
`User-agent: *`
`Allow: /`
	Deux robots : YandexBot [.ru] et
	SemrushBot [.us] ne pourront pas
	explorer le répertoire /private/
	ni la page /members/users.html
	en revanche,
	Tous les autres robots seront
	autorisés à fouiller tout à partir de /

Dans le dernier exemple, après les interdictions pour ces deux robots, l'autorisation pour les autres (`User-agent: *` et `Allow: /`) est superflue car tout ce qui n'est pas interdit est autorisé. C'est juste pour faire exemple de deux règles dans le même fichier `robots.txt` (toute règle commence par un ou plusieurs `User-agent`).

Documentation : https ://datatracker.ietf.org/doc/html/draft-rep-wg-topic-00

1.2 Mode de fonctionnement de Robots Class

Les robots fouillent les sites web à la recherche des fichiers `robots.txt` à la racine de chaque site web pour les lire. Si, à la place d'un fichier `robots.txt` vous créez un répertoire `robots.txt`, celui-ci sera lu et le fichier qui sera envoyé au demandeur (le robot) sera le fichier `index.php`. En fait, pour traquer les traqueurs, vous les trompez en leur donnant l'illusion de lire un fichier `robots.txt` alors qu'ils liront un répertoire qui enverra son fichier `index.php`.

Le fichier `index.php` contient la classe **Robots Class** ainsi qu'une instanciation qui provoque l'enregistrement et la réponse au robot, c'est tout.

1.3 Mise en œuvre

Ouvrez le fichier `index.php` avec un éditeur de texte et modifiez les membres de la classe en fonction de vos propres informations et choix.

Au paragraphe « *How to record the visit ?* », membre `$rectype`, renseignez votre choix :

- `none` si vous ne voulez rien faire ;
- `database` si vous souhaitez l'enregistrement dans une base de données (vous aurez alors à renseigner le paragraphe « *For database recording* ») ;
- `file:robots.log` pour enregistrer dans un fichier `robots.log` dans le même répertoire (`/robots.txt/robots.log`). Vous pouvez changer le nom du fichier mais rien ne l'impose, vous pouvez garder celui qui vous est proposé. Si vous effectuez ce choix, assurez-vous que le répertoire `robots.txt` ait les droits en écriture (`chmod 777 robots.txt`) ;
- `mail:your.address@domain.tld` pour envoyer une notification de la visite à l'adresse mail indiquée. Bien que ce choix soit possible, je vous suggère de ne pas choisir ce mode (sauf pour des tests) car votre boîte à lettres risque d'être rapidement encombrée.

Membre `$website` : Si vous n'avez qu'un seul site, ce champ peut être vide (`private $website = "";`). Inutile en effet de mentionner à chaque fois dans la base de données le nom du site s'il n'y en a qu'un. On le connaît et ça prend de la place dans la base de données (si `database` est le choix de l'enregistrement). Si vous avez plusieurs sites, c'est intéressant de le mentionner (ex : `private $website = "www.mywebsite.com";`) dans chacun des fichiers `index.php` que vous placerez dans le répertoire `robots.txt` de chaque hôte (virtuel ou non). Le nom du site sera enregistré dans la base de données ou inscrit dans le mail mais pas dans le fichier journal (`robots.log`) de

chacun des sites.

Si vous souhaitez enregistrer les visites dans une base de données (ce que je vous conseille pour regrouper toutes les visites et qui permet de faire des statistiques plus facilement que par mail ou fichiers texte), renseignez le paragraphe « *For database recording* », notamment les membres suivants :

- `$server` Le numéro IP du serveur de base de données ou son FQDN ;
- `$base` Le nom de la base de données ;
- `$user` Le nom de l'utilisateur de cette base de données ;
- `$pass` Son mot de passe ;
- `$table` Le nom de la table SQL (que vous pouvez changer mais je vous propose de le conserver) ;
- `$mydbtype` le type de base de données (`0` = mysql ; `1` = pgsql).

A la fin de chaque action (ne rien faire, enregistrer dans un fichier, dans une base de données ou notifier par mail) le processus s'arrête en fournissant un contenu de fichier `robots.txt` que le webcrawler est supposé trouver.

Si vous autorisez les indexations de tous vos fichiers et répertoires, le fichier `robots.txt` peut être absent ou vide. Dans ce cas, le membre `$botfile` peut être vide ou null (`private $botfile = "";`)

Si vous souhaitez placer des règles vous DEVEZ les écrire sur une SEULE et même ligne, chaque ligne du fichier `robots.txt` se terminant par un retour à la ligne (`\n`). Pour le premier exemple de fichier `robots.txt` donné ci-dessus, on écrirait :

```
private $botfile = "User-agent: *\nDisable: /\n";
```

Pour le second, on écrirait :

```
private $botfile = "User-agent: Googlebot\nDisable: /private/\n";
```

Vous pouvez aussi, si vous le désirez vous créer un fichier `robots.txt` et le déclarer au membre `$rfile` (exemple : `private $rfile = "myrobots.txt";`). S'il est null (`$rfile = "";`) ou s'il est mentionné mais absent, c'est le membre `$botfile` (null ou renseigné) qui sera envoyé au robot.

Après avoir effectué le suivi comme vous le souhaitez (rien, fichier, base de données, mail), le processus s'arrête en fournissant le fichier `robots.txt` avec : `die($this->botfile);` (membre `$botfile`) ou le contenu d'un réel fichier spécifique désigné par `$rfile`.

Vous pouvez vérifier la production du fichier `robots.txt` en mettant `$rectype` à `none` (pour ne rien enregistrer) et taper sur la ligne de commande : `php index.php` ; vous remettrez ensuite `$rectype` à la valeur que vous souhaitez.

C'est tout.

Vous pouvez aussi modifier le code du script **Robots Class** pour accepter l'ajout de nouvelles informations en consultant celles que vous pouvez obtenir en exécutant ce petit script :

```
<pre>
<?php $headers = getallheaders(); print_r($headers); ?>
</pre>
```

Fichiers joints au package :

— `robots.sql` script SQL pour créer votre table SQL Robots ;
— `visits_base.txt` Exemple d'enregistrements de plusieurs sites (dans une base de données) ;
— `visits_file.txt` Exemple d'enregistrements (dans un fichier).

Organisation pour des hôtes virtuels sur le même hébergement (avec le serveur web apache2) :

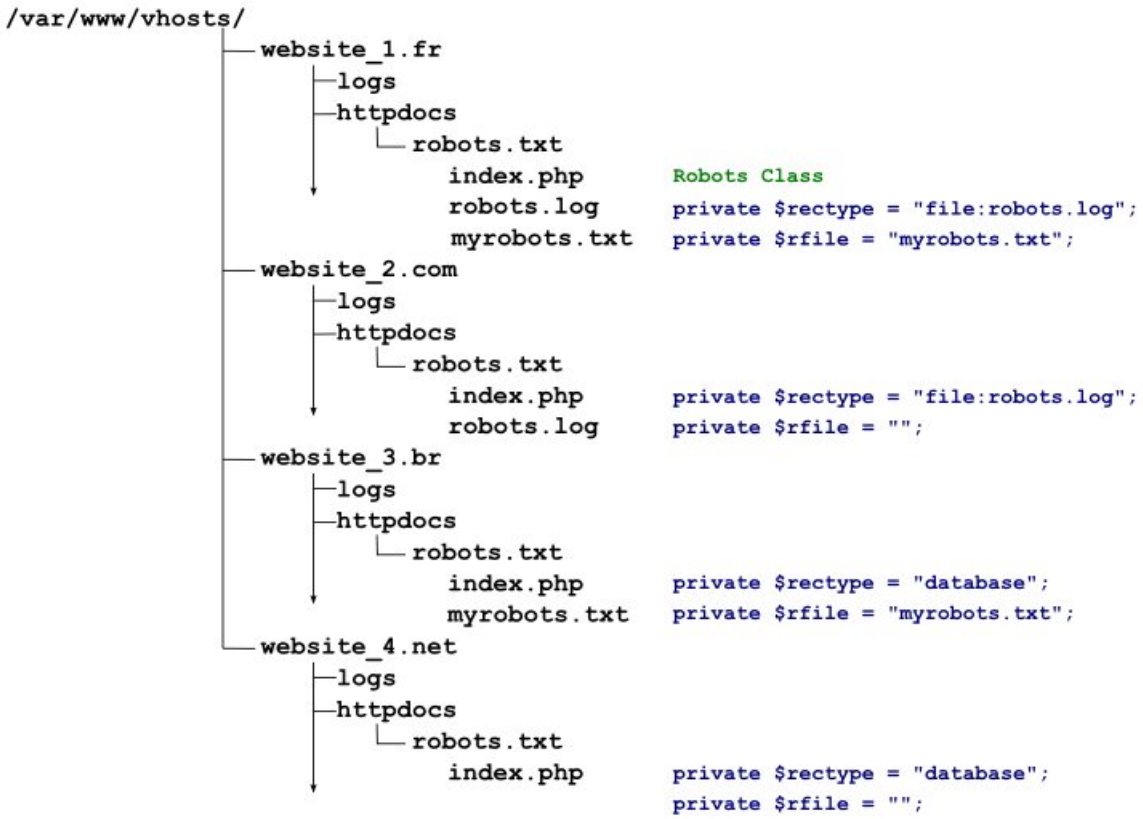

FIGURE 1 – Hébergement avec hôtes virtuels (plusieurs sites)

2 English documentation

2.1 The `robots.txt` file

If present, this text file must be located in the root directory of the website. If it is absent or empty, the robots will have access to all the files and directories of the website to index them. If you do not want a page or directory to be crawled by robots, you will have to write the rules you want in a `robots.txt` file.

The rules include first the designation of the robot (`User-agent`) and the permissions or prohibitions (`Allow`, `Disallow`). In the designation of the robot, the asterisk means « all robots » and the permissions or prohibitions start at the root directory of the site.

`User-agent: *` `Disallow: /`	All robots aren't allowed to crawl the site from the root directory
`User-agent: Googlebot` `Disallow: /private/`	Googlebot should not crawl the /private/ directory

`User-agent: YandexBot`
`User-agent: SemrushBot`
`Disallow: /private/`
`Disallow: /members/users.html`
 	
`User-agent: *`
`Allow: /`
	Two bots : YandexBot [.ru] and
	SemrushBot [.us]
	will not access to the /private/ directory
	neither the page /members/users.html
	on the other hand,
	All other bots will be
	allowed to dig everything from the root

In the last example, after the prohibitions for two robots, the authorization for the others (`User-agent: *` and `Allow: /`) is superfluous because everything not prohibited is authorized. It's just to make an example of two rules in the same `robots.txt` file (every rule starts with one or more `User-agent`).

Documentation : https ://datatracker.ietf.org/doc/html/draft-rep-wg-topic-00

2.2 Operating mode of Robots Class

Robots search websites for `robots.txt` files at the root directory of each website to read them. If, instead of a `robots.txt` file, you create a

`robots.txt` directory, it will be read and the file that will be sent to the requester (the robot) will be the `index.php` file. In fact, to track trackers, you trick them into giving them the illusion of reading a `robots.txt` file while they will be reading a directory which will send its `index.php` file (the **Robots Class**).

The `index.php` file contains the class **Robots Class** and an instantiation which causes the registration and the answer to the robot, that's all.

2.3 Implementation

Open the `index.php` file with a text editor and modify the class members according to your own informations and choices.

In the paragraph « *How to record the visit ?* »", member `$rectype`, write your choice :

- `none` if you don't want to do anything ;
- `database` if you want the recording in a database (you will then have to fill in the paragraph « *For database recording* ») ;
- `file:robots.log` to save to a `robots.log` file in the same directory (`/robots.txt/robots.log`). You can change the name of the file but nothing imposes it, you can keep the one that is proposed. If you make this choice, be sure that the `robots.txt` directory has write rights (`chmod 777 robots.txt`) ;
- `mail:your.address@domain.tld` to send a notification of the visit to the specified email address. Although this choice is possible, I suggest that you do not choose this mode (except for tests) because your mailbox may quickly become cluttered.

Member `$website` : If you only have one site, this field can be empty (`private $website = "";`). There is no need to mention the name of the site each time in the database if there is only one. We know it and it takes space in the database (if `database` is the choice for the record). If you have several sites, it is interesting to mention it (ex : `private $website = "www.mywebsite.com";`) in each of the `index.php` files that you will place in the `robots.txt` directory of each virtual host. The name of the site will be recorded into the database or written in the email but not in the log file (`robots.log`) of each of the sites.

If you want to record the visits in a database (which I recommend to group all the visits and which allows you to make statistics more easily than by mail or text files), fill in the paragraph « *For database recording* », in particular the following members :

— `$server` The IP number of the database server or its FQDN ;
— `$base` The name of the database ;
— `$user` The name of the user of this database ;
— `$pass` His password ;
— `$table` The name of the SQL table (which you can change but I suggest you keep it) ;
— `$mydbtype` the database type (`0` = mysql ; `1` = pgsql).

At the end of each action (do nothing, save to a file, to a database or notify by email) the process stops by providing a `robots.txt` file content that the webcrawler is supposed to find.

If you allow indexing of all your files and directories, the `robots.txt` file may be missing or empty. In this case, the `$botfile` member can be empty or null (`private $botfile = "";`)

If you want to place rules you MUST write them on a SINGLE line, each file line ending with a newline (\n). For the first example `robots.txt` file given above as example, we would write :

```
private $botfile = "User-agent: *\nDisable: /\n";
```

For the second, we would write :

```
private $botfile = "User-agent: Googlebot\nDisable: /private/\n";
```

You can also, if you wish, create a file `robots.txt` and declare it to the member `$rfile` (example : `private $rfile = "myrobots.txt";`). It will be sent to the robot. If it is null (`$rfile = "";`) or if it is mentioned but absent, the value of the `$botfile` member (null or filled in) will be sent to the robot.

After tracking as desired (nothing, file, database, mail), the process terminates providing the `robots.txt` file with : `die($this->botfile);` (member `$botfile`) or the content of a real specific file mentionned by `$rfile`

You can check the production of the `robots.txt` file by setting `$rectype` to `none` (to save nothing) and typing on the command line : `php index.php` ; you will then set `$rectype` back to whatever value you want.

That's all.

You can also modify the code of the script **Robots Class** to accept the addition of new information by looking at what you can get by running this little script :

```
<pre>
<?php $headers = getallheaders(); print_r($headers); ?>
</pre>
```

Files attached to the package :

— `robots.sql` an SQL script to create your Robots SQL table ;
— `visits_base.txt` an example of records into a database for several websites ;
— `visits_file.txt` an example of records into a file.

Organization for virtual hosts on the same hosting (with apache2) :

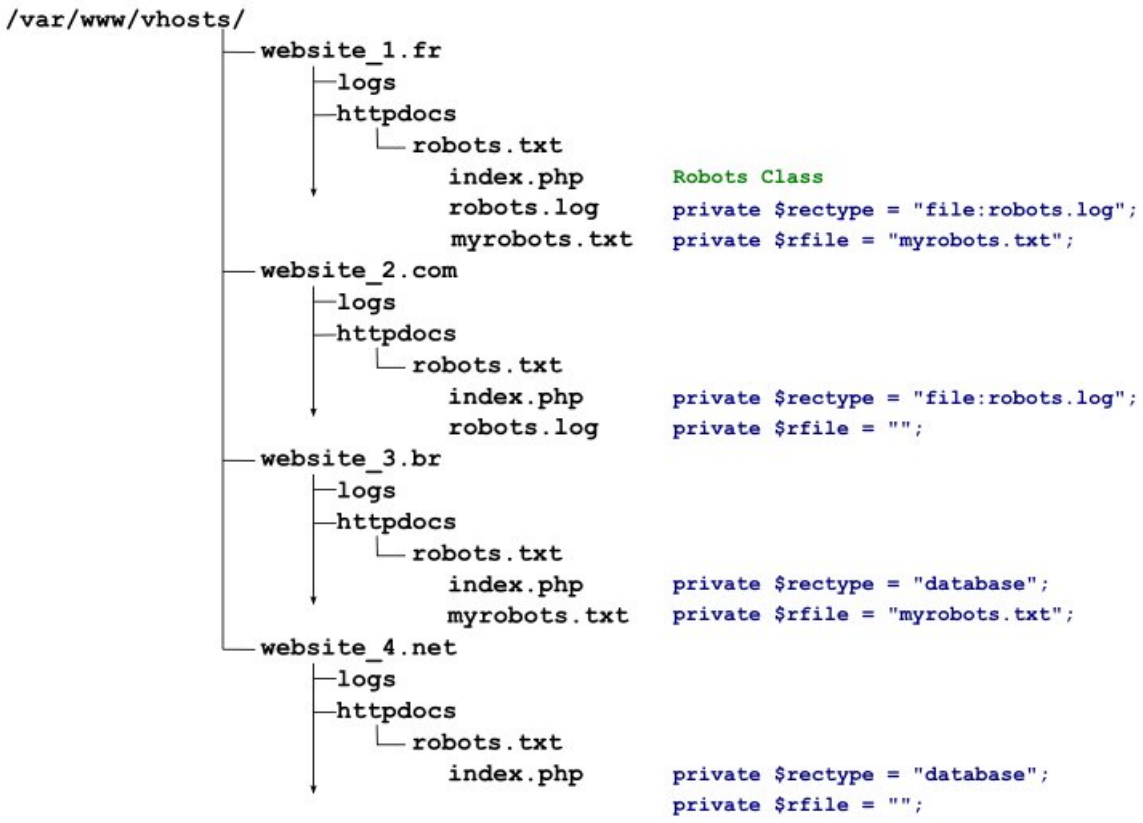

FIGURE 2 – Hosting with virtual hosts (multiple sites)

Robots Class v1.1

February 28, 2022 Pierre FAUQUE, *<pierre@fauque.net>*

Document powered by LaTeX