Web-Server: robots.txt

Sinn der `robots.txt`

Viele Robots/Spider/Crawler/Bots durchsuchen wahllos das Internet nach Inhalten.
Manchmal ist es sinnvoll, bestimmte Seiten oder Bereiche der eigenen Site vor diesen Robots zu schützen.

Beispiele:

- Seiten die sich täglich ändern oder an denen noch gearbeitet wird.
- Programmdateien oder Logfiles.
- Bilder und Download-Dateien.

Aufbau der `robots.txt`

Eine Textdatei mit dem Namen `robots.txt` wird im Stammverzeichnis des Webs angelegt.

Das Schema der `robots.txt` Datei ist einfach aufgebaut:

- Welchen Robot betrifft es?
  `User-Agent`
- Was darf er, was nicht?
  `Allow` und `Disallow`
- Nächster Regelblock wird wieder mit `User-Agent` eingeleitet.
- Kommentare beginnen mit `#` und gehen bis zum Ende der Zeile.

Syntax von `User-Agent`:

```
User-Agent: [Name|*]
```

Entweder die Regel gilt nur einen Robot, oder mit `*` für alle.

Syntax von `Allow`:

```
Allow: [Datei|Pfad&|*]
```

Ein Pfad beginnt und endet immer mit einem `/` (slash).

Da die Allow Anweisung aber erst später als der eigentliche Standard eingeführt wurde, wird sie noch nicht von allen Robots unterstützt. Von daher sollte man sich nicht darauf verlassen und lieber nur Disallow benutzen.

Syntax von `Disallow`:

```
Disallow: [Pfad]
```

Bei Disallow ist keine Wildcard `*` erlaubt.

Hier muß auch aktiv auf die Pfade geachtet werden. Ein `Disallow /index` würde evtl. die Datei `index.html` und das Verzeichnis `index-dateien/` ausschließen.

Beispiele für `robots.txt`

Es ist auf jeden Fall sinnvoll, eine minimale `robots.txt` zu erstellen:

```
# robots.txt for http://www.domain.tld/

# Zugriff auf alle Dateien erlauben

User-agent: *

Disallow:
```

```
# robots.txt fuer http://www.domain.tld/
```

```
User-agent: BeispielRobot

Disallow: /temp/       # Die Dateien sind sehr kurzlebig

Disallow: /logfiles/   # Die ändern sich jeden Tag

Disallow: /bilder/     # Bilder nicht downloaden

Disallow: /cgi-bin/    # CGI Ausgaben nicht indexieren

Disallow: /news.html   # Die news.html ändert sich täglich
```

Mit `User-agent: BeispielRobot` bestimmt man, daß die Anweisungen nur für den Crawler `BeispielRobot` gelten.

Mit den einzelnen `Disallow` Einträgen bestimmt man Dateien und Verzeichnisse die nicht indexiert werden sollen.

Um alle Crawler aus einem Verzeichnis rauszuhalten, benutzt man den Wildchar `*`.

```
# Alle Robots ausschließen

User-agent: *

Disallow: /temp/
```

Wenn man nicht gleich alle Crawler, sondern nur ein paar bestimmte meint, kann man diese so angeben:

```
# Massendownloader vom CGI Verzeichnis fernhalten

User-agent: wget

User-agent: webzip

User-agent: webmirror
```

```
User-agent: webcopy

Disallow: /cgi-bin/
```

Um seine Site ganz von der Indexierung auszuschließen kann man folgendes benutzen:

```
# Ganze Site für alle Robots sperren

User-agent: *

Disallow: /
```

Wenn man den Slash (/) wegläßt, gibt man Seine Site ganz für die Indexierung frei.

```
# Ganze Site freigeben

User-agent: *

Disallow:
```

Ein Working Draft der IETF führt neben der Disallow Anweisung auch die Allow Anweisung ein:

User-agent: *
Disallow: /temp/
Allow: /temp/daily.html

Fehler vermeiden & Syntax prüfen

- Häufiger Fehler: Ein HTML-Editor wird beim erstellen der `robots.txt` benutzt und fügt ungewollten HTML-Code ein.
- Beim FTP-Transfer sollte die `robots.txt` im ASCII Modus übertragen werden.

Web-Server: robots.txt

- Der Dateiname besteht nur aus Kleinbuchstaben!
- Man sollte immer einen Syntax-Check machen um seine `robots.txt` zu prüfen. (Link siehe unten.)

Grenzen der `robots.txt`

- Man kann mit der `robots.txt` keine Dateien vor Zugriffen schützen.
- Kein Robot ist verbindlich verpflichtet sich an die `robots.txt` zu halten.
- Mit der `robots.txt` kann man keinen Robot dazu bringen etwas bestimmtes zu indexieren.

Links zur `robots.txt`

- IETF Working Draft: A Method for Web Robots Control
- robots.txt Syntax Checker
- Artikel bei Suchfibel : Draussen bleiben! robots.txt

Eindeutige ID: #1004
huschi
2008-11-10 13:35