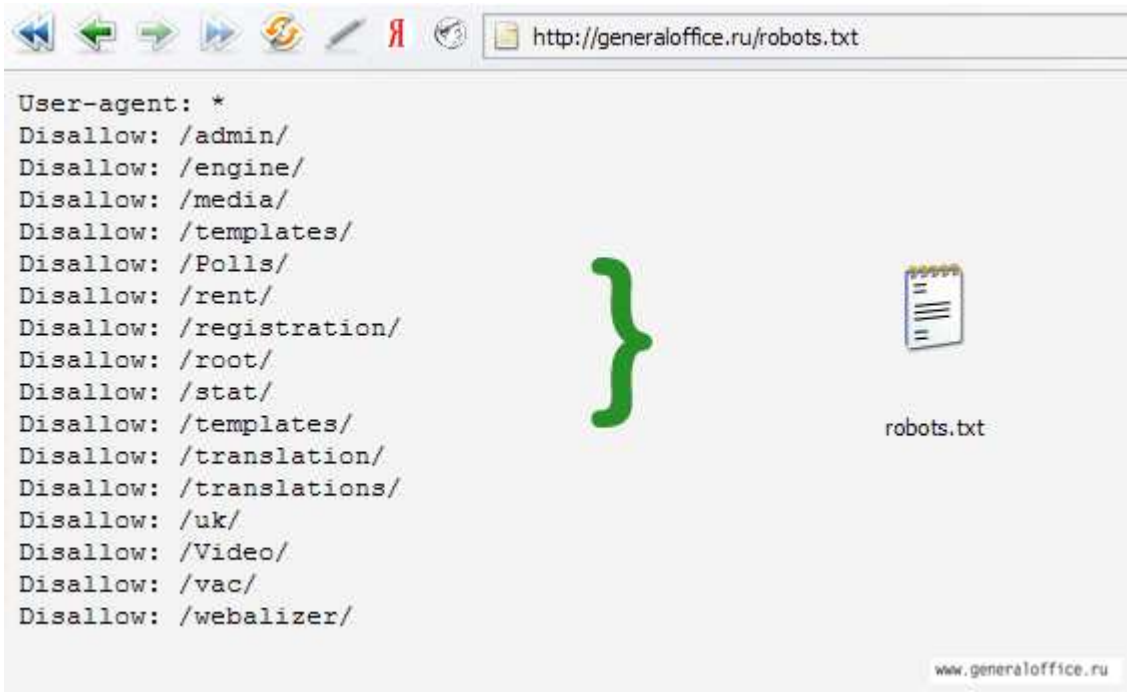
Правильный файл robots.txt

http://robotstxt.org.ru/ - Все о файле robots.txt по-русски

robots.txt — файл ограничения доступа к содержимому роботам на http-сервере. Файл должен находиться в корне сайта (т.е. иметь путь относительно имени сайта /robots.txt). Использование файла добровольно, стандарт был принят консорциумом 30 января 1994 года в списке рассылки robots-request@nexor.co.uk и с тех пор используется большинством известных поисковых машин.



Файл robots.txt используется для частичного управления индексированием сайта поисковыми роботами. Этот файл состоит из набора инструкций для поисковых машин, при помощи которых можно задать области сайта, которые не должны индексироваться.

Файл robot.txt

Поисковые сервера всегда перед индексацией вашего ресурса ищут в корневом каталоге вашего домена файл с именем "robots.txt" (http://www.mydomain.com/robots.txt). Этот файл сообщает роботам (паукам-индексаторам), какие файлы они могут индексировать, а какие нет.
Формат файла robots.txt - особый. Он состоит из записей. Каждая запись состоит из двух полей: строки с названием клиентского приложения (user-agent), и одной или нескольких строк, начинающихся с директивы Disallow:

<Поле> ":" <значение>

Robots.txt должен создаваться в текстовом формате Unix. Большинство хороших текстовых редакторов уже умеют превращать символы перевода строки Windows в Unix. Либо ваш FTP-клиент должен уметь это делать. Для редактирования не пытайтесь пользоваться HTML-редактором, особенно таким, который не имеет текстового режима отображения кода.

Поле User-agent

Строка User-agent содержит название робота. Например:

User-agent: googlebot

Если вы обращаетесь ко всем роботам, вы можете использовать символ подстановки "*":

User-agent: *

Названия роботов вы можете найти в логах вашего веб-сервера. Для этого выберите только запросы к файлу robots.txt. большинство поисковых серверов присваивают короткие имена своим паукам-индексаторам.

Поле Disallow:

Вторая часть записи состоит из строк Disallow. Эти строки - директивы для данного робота. Они сообщают роботу какие файлы и/или каталоги роботу неразрешено индексировать. Например следующая директива запрещает паукам индексировать файл email.htm:

Disallow: email.htm

Директива может содержать и название каталога:

Disallow: /cgi-bin/

Эта директива запрещает паукам-индексаторам лезть в каталог "cgi-bin".

В директивах Disallow могут также использоваться и символы подстановки. Стандарт диктует, что директива /bob запретит паукам индексировать и /bob.html и /bob/index.html.

Если директива Disallow будет пустой, это значит, что робот может индексировать ВСЕ файлы. Как минимум одна директива Disallow должна присутствовать для каждого поля User-agent, чтобы robots.txt считался верным. Полностью пустой robots.txt означает то же самое, как если бы его не было вообще.

Пробелы и комментарии

Любая строка в robots.txt, начинающаяся с #, считается комментарием. Стандарт разрешает использовать комментарии в конце строк с директивами, но это считается плохим стилем:

Disallow: bob #comment

Некоторые пауки не смогут правильно разобрать данную строку и вместо этого поймут ее как запрет на индексацию ресурсов bob#comment. Мораль такова, что комментарии должны быть на отдельной строке.

Пробел в начале строки разрешается, но не рекомендуется.

Disallow: bob #comment

Примеры

Следующая директива разрешает всем роботам индексировать все ресурсы сайта, так как используется символ подстановки "*".

User-agent: *
Disallow:

Эта директива запрещает всем роботам это делать:

User-agent: *
Disallow: /

Данная директива запрещает всем роботам заходить в каталоги "cgi-bin" и "images":

User-agent: *
Disallow: /cgi-bin/
Disallow: /images/

Данная директива запрещает роботу Roverdog индексировать все файлы сервера:

User-agent: Roverdog
Disallow: /

Данная директива запрещает роботу googlebot индексировать файл cheese.htm:

User-agent: googlebot
Disallow: cheese.htm

Если вас интересуют более сложные примеры, попутайтесь вытянуть файл robots.txt с какого-нибудь крупного сайта, например CNN или Looksmart.

Дополнения к стандартам
Несмотря на то, что были предложения по расширению стандарта и введению директивы Allow или учета версии робота, эти предложения формально так и не были утверждены.

Поход в поисках robots.txt

При проверке нашего валидатора robots.txt (см. конец статьи), нам понадобилось найти много-много "корма" для него. Мы создали спайдер, который скачивал с каждого найденного сайта лишь один файл robots.txt. Мы прошлись по всем ссылкам и доменам, занесенным в Open Directory Project. Так мы прошлись по 2.4 миллионам URL и накопали файлов robots.txt примерно на 75 килобайт.

Во время этого похода мы обнаружили огромное количество проблем с файлами robots.txt. Мы увидели, что 5% robots.txt плохой стиль, а 2% фалов были настолько плохо написаны, что ни один робот не смог бы их понять. Вот список некоторых проблем, обнаруженных нами:

Перевернутый синтаксис

Одна из самых распространенных ошибок - перевернутый синтаксис:

User-agent: *
Disallow: scooter

А должно быть так:

User-agent: scooter
Disallow: *

Несколько директив Disallow в одной строке:

Многие указывали несколько директив на одной строке:

Disallow: /css/ /cgi-bin/ /images/

Различные пауки поймут эту директиву по разному. Некоторые проигнорируют пробелы и поймут директиву как запрет на индексацию каталога /css//cgi-bin//images/. Либо они возьмут только один каталог (/images/ или /css/) и проигнорируют все остальное.

Правильный синтаксис таков:

Disallow: /css/
Disallow: /cgi-bin/
Disallow: /images/

Перевод строки в формате DOS:

Еще одна распространенная ошибка - редактирование файла robots.txt в формате DOS. Несмотря на то, что из-за распространенности данной ошибки многие пауки-идексаторы научились понимать ее, мы считаем это ошибкой. Всегда редактируйте свой robots.txt в режиме UNIX и закачивайте файл на сайт в режиме ASCII. Многие FTP-клиенты умеют при закачке в текстовом режиме переводить символы строки из DOS-формата в UNIX-формат. Но некоторые этого не делают.

Комментарии в конце строки:

Согласно стандарту, это верно:

Disallow: /cgi-bin/ #this bans robots from our cgi-bin

Но в недавнем прошлом были роботы, которые заглатывали всю строку в качестве директивы. Сейчас нам такие роботы неизвестны, но оправдан ли риск? Размещайте комментарии на отдельной строке.

Пробелы в начале строки:

Disallow: /cgi-bin/

Стандарт ничего не говорит по поводу пробелов, но это считается плохим стилем. И опять-таки, стоит ли рисковать?

Редирект на другую страницу при ошибке 404:

Весьма распространено, когда веб-сервер при ошибке 404 (Файл не найден) выдает клиенту особую страницу. При этом веб-сервер не выдает клиенту код ошибки и даже не делает редиректа. В этом случае робот не понимает, что файл robots.txt отсутствует, вместо этого он получит html-страницу с каким-то сообщением. Конечно никаких проблем здесь возникнуть не должно, но стоит ли рисковать? Бог знает, как разберет робот этот html-файл, приняв его за robots.txt. чтобы этого не происходило, поместите хотя бы пустой robots.txt в корневой каталог вашего веб-сервера.

Конфликты директив:

Чтобы вы сделали на месте робота slurp, увидев данные директивы?

```
User-agent: *
Disallow: /
#
User-agent: slurp
Disallow:
```

Первая директива запрещает всем роботам индексировать сайт, но вторая директива разрешает роботу slurp это делать. Так что же все-таки должен делать slurp? Мы не можем гарантировать, что все роботы поймут эти директивы правильно. В данном примере slurp должен проиндексировать весь сайт, а все остальные не должны уйти прямо с порога.

Верхний регистр всех букв - плохой стиль:

```
USER-AGENT: EXCITE
DISALLOW:
```

Несмотря на то, что стандарт безразлично относится к регистру букв в robots.txt, в именах каталогов и файлов регистр все-таки важен. Лучше всего следовать примерам и в верхнем регистре писать первые буквы только в словах User и Disallow.

Список всех файлов

Еще одна ошибка - перечисление всех файлов в каталоге:

```
Disallow: /AL/Alabama.html
Disallow: /AL/AR.html
Disallow: /Az/AZ.html
Disallow: /Az/bali.html
Disallow: /Az/bed-breakfast.html
```

Вышеприведенный пример можно заменить на:

```
Disallow: /AL
Disallow: /Az
```

Помните, что начальная наклонная черта обозначает, что речь идет о каталоге. Конечно, ничто не запрещает вам перечислить парочку файлов, но мы речь ведем о стиле. Данный пример взят из файла robots.txt, размер которого превышал 400 килобайт, в нем было упомянуто 4000 файлов! Интересно, сколько роботов-пауков, посмотрев на этот файл, решили больше не приходить на этот сайт.

Есть только директива Disallow!

Нет такой директивы Allow, есть только Disallow. Этот пример неверный:

```
User-agent: Spot
Disallow: /john/
allow: /jane/
```

Правильно будет так:

```
User-agent: Spot
Disallow: /john/
Disallow:
```

Нет открывающей наклонной черты:

Что должен сделать робот-паук с данной директивой:

```
User-agent: Spot
Disallow: john
```

Согласно стандартам эта директива запрещает индексировать файл "john" и каталог john". Но лучше всего, для верности, использовать наклонную черту, чтобы робот мог отличить файл от каталога.

Еще мы видели, как люди записывали в файл robots.txt ключевые слова для своего сайта (подумать только - для чего?).

Бывали такие файлы robots.txt, которые были сделаны в виде html-документов. Помните, во FrontPage делать robots.txt не стоит.

Неправильно настроенный сервер

Почему вдруг на запрос robots.txt веб-сервер выдает бинарный файл? Это происходит в том случае, если ваш веб-сервер настроен неправильно, либо вы неправильно закачали на сервер сам файл.

Всегда после того, как вы закачали файл robots.txt на сервер, проверяйте его. Достаточно в броузере набрать простой запрос:

http://www.mydomain.com/robots.txt

Вот и все что нужно для проверки.

Особенности Google:

Google - первый поисковый сервер, который поддерживает в директивах регулярные выражения. Что позволяет запрещать индексацию файлов по их расширениям.

User-agent: googlebot
Disallow: *.cgi

В поле user-agent вам следует использовать имя "googlebot". Не рискуйте давать подобную директиву другим роботам-паукам.

META-тег robots
META тег robots служит для того, чтобы разрешать или запрещать роботам, приходящим на сайт, индексировать данную страницу. Кроме того, этот тег предназначен для того, чтобы предлагать роботам пройтись по всем страницам сайта и проиндексировать их. Сейчас этот тег приобретает все большее значение.

Кроме того, этим тегом могут воспользоваться те, кто не может доступиться к корневому каталогу сервера и изменить файл robots.txt.

Некоторые поисковые сервера, такие как Inktomi например, полностью понимают мета-тег robots. Inktomi пройдет по всем страницам сайта если значение данного тега будет "index,follow".

Формат мета-тега Robots

Мета тег robots помещается в тег html-документа. Формат достаточно прост (регистр букв значения не играет):

<HTML>
<HEAD>
<META NAME=ROBOTS" CONTENT="NOINDEX, NOFOLLOW">
<META NAME="DESCRIPTION" CONTENT="Эта страница ….">
<TITLE>...</TITLE>
</HEAD>
<BODY>

Значения мета-тега robots

Данному мета-тегу можно присвоить варианта четыре значений. Атрибут content может содержать следующие значения:

index, noindex, follow, nofollow

Если значений несколько, они разделяются запятыми.

В настоящее время лишь следующие значения важны:

Директива INDEX говорит роботу, что данную страницу можно индексировать.

Директива FOLLOW сообщает роботу, что ему разрешается пройтись по ссылкам, присутствующим на данной странице. Некоторые авторы утверждают, что при отсутствии данных значений, поисковые сервера по умолчанию действуют так, как если бы им даны были директивы INDEX и FOLLOW. К сожалению это не так по отношению к поисковому серверу Inktomi. Для Inktomi значения по умолчанию равны "index, nofollow".

Итак, глобальные директивы выглядят так:

Индексировать всё = INDEX, FOLLOW

Не индексировать ничего = NOINDEX,NOFLLOW

Примеры мета-тега robots:

<META NAME=ROBOTS" CONTENT="NOINDEX, FOLLOW">
<META NAME=ROBOTS" CONTENT="INDEX, NOFOLLOW">
<META NAME=ROBOTS" CONTENT="NOINDEX, NOFOLLOW">