# Aura proff manual

http://www.auraproff.biz/
mailto: aura.proff@gmail.com

Содержание:

# I. Описание

*Модули парсинга*

### Google

*http://www.google.com*

1000 результатов с запроса, 100 страниц на выдаче,
использование разных data центров.

### Yahoo

*http://search.yahoo.com*

1000 результатов с запроса, 10 страниц на выдаче.

### Explorer

*http://siteexplorer.yahoo.com*

1000 результатов с запроса, 100 страниц на
выдаче; операторы: link, linkdomain, site.

### Live

*http://search.live.com*

1000 результатов с запроса, 200 страниц на выдаче.

*Дополнительные модули парсинга*

### Snipets

Парсинг 100 первых снипетов google от каждого запроса;

*Все модули поддерживают:*

- многопоточность;
- парсинг с прокси, задержкой;
- маскировка под сотни браузеров;
- gzip сжатие (увеличивает скорость и снижает потребление трафика);
- все операторы языка запросов для каждого поисковика;
- возможность использовать любые языки для составления запросов;
- удаление дубликатов по домену и по строке;
- полное логирование всех действий.

*Модули фильтрации:*

### Regexp match

Выборка url'ов по регулярным выражениям, возможность замены подстроки.

### Ping

Отбирает url'ы по статусу ответа 200, 301, 302 и т.д.

### PR

Производит выборку url'ов по google page rank.

### Scan

Сканирует ресурсы на предмет вхождения искомой строки (регулярного выражения).

*Все модули поддерживают:*

- многопоточность (для некоторых не актуальна);
- полное логирование всех действий.

*Прочее:*

- Четкое разделение заданий: парсинг и фильтрация.
- Система запросов и подзапросов (доп. запросов).
- Легкое внедрение вариаторов - им может быть любой фаил.
- Система профайлов.
- Средство для роботы с файлами;
- В скрипт встроено средство для роботы с прокси; возможности:
  - скачивание прокси с url'ов;
  - многопоточная проверка на активность - пинг;
  - запуск вручную и через *cron*;

## II. Требования

- *unix* подобная ос - *linux, freebsd, mac os* и т.д. (с доступными командами: *ps, grep, awk*);
- *php 4.x* или *5.x* с необходимыми директивами *php.ini* (проверяются скриптом);
- *python*, желательно последней версии - *2.5.1*;
- *apache 1.3.x* или *2.x.x*;
- браузер для управления (рекомендуется *firefox, opera, safari*).

## III. Установка

1. Скачайте архив с программой из вашего аккаунта.
2. Поместить папку со скриптом на сервер, где он будет доступен через http.
3. Установить следующие права:
   - корневая папка: *0755* или *0777*;
   - файлы python (**.py, которые в корне*): *0755*;
   - на *все* остальные папки и файлы во *всех* папках, т.е *рекурсивно: 0777.*
4. Удобнее всего это сделать через *ssh* (порядок команд сохранить):

   ```
   chmod -R 0777 distr
   chmod 0755 distr/*.py
   chmod 0755 distr
   ```
5. Запустить через браузер *http://yourdomain.com/distr/install.php*
6. Скрипт проверит систему по требованиям, проверит права - *permissions;* если проблем нет продолжите установку, следуя указаниям, где вам надо будет ввести ключ лицензии, пароль на панель и т.д.
7. Поставить на *ежеминутный запуск* по крону *файл starter.py;* пример команды предложит установщик.
8. Сборщик прокси лучше поставить на крон, в разделе Proxy есть пример команды для вашего сервера, так же установишик предложит команду.
9. После установки следует *удалить* или *сделать недоступны м* из веба *install.php*; *обратите внимание* на настройки пути к интерпритатору питона - *where is python* и

архиватору zip - *where is zip*, на некоторых ос установщик может предложить неверный путь.

После установки, скрипт “привязывается” к ip сервера - в зоне для клиента он отображается, так же как и время последней установки. Для смены ip просто запустите инсталятор там где хотите работать и скрипт сам “перепривяжется”.

## IV. Парсинг

Все необходимое находится в одноименном разделе - Parsing.

*Добавление нового задания:*

1. *Add new task*
   Выбираете что парсить - *type*:
   - *google* - google.com
   - *yahoo* - search.yahoo.com
   - *explorer* - siteexplorer.yahoo.com
   - *live* - search.live.com
   - *snipets* - парсер снипетов с google.com
2. Профайл настроек - *profile* и количество потоков - *threads*.
3. Файл для результатов - *output*.
4. В следующем шаге вводятся запросы - *queries* и дополнительные запросы - *subqueries*. Все указывается в столбик, т.е разделяется символом переноса строки \n.

*Принцип работы:* например в queries есть слово blog, а в subqueries site:com и site:net, это приведет к парсингу трех запросов: blog, blog site:com, blog site:net.
При использовании операторов убедитесь, что они действуют в тех поисковиках, которые используете, например, в search.live.com нет оператора inurl.
Модуль *explorer* поддерживает *три и только три типа запросов:*

- *link:www.someurl.com*
- *site:www.someurl.com*
- *linkdomain:www.someurl.com*

*Например:* site:google.com, linkdomain:adobe.com, link:umaxforum.com/forumdisplay.php
Если у вас есть файл с запросами, который есть в files, его можно вставить *макросом:* *{somefile.txt}*. Доступные вариаторы отображены справа при добавлении запросов.
*Парсинг "бэков"* с помощью siteexplorer.yahoo.com, например, удобнее всего сделать так: В queries написать link: а в subqueries написать {somefile.txt}, где somefile.txt - файл с url'ами.
После всех манипуляций нажмите *Add*.

5. Все задание создано, для запуска нажмите *Start*, для остановки и удаления - *Stop* и *Delete*. Исполняемое задание подсвечено, подробные логи можно посмотреть, нажав на процент прогресса, *в случае проблем всегда сначала смотрите логи.* Дубликаты удаляются в конце парсинга.
6. Результаты накапливаются постепенно в *output* файл; все модули кроме модуля *snipets* складывают туда url’ы, модуль *snipets* собирает снипеты (короткое описание результата выдачи у google) по одному на строку.

*В заключении:*
При длительном многопоточном парсинге *для всех* поисковиков нужны прокси, просто у разных поисковиков разная толерантность :) . Например, google банит быстро но и

разбанивает быстро, в отличии от yahoo, а live.com на одном потоке без задержки и без прокси позволил мне выпарсить 500к ресурсов. Нет смысла парсить при одном потоке и несколько k полуживых прокси, как и без прокси в несколько потоков. Скорость так же завивисит от поисковика, скажем yahoo медленнее всех, т.к. 10 результатов на страницу.

## V. Фильтрация

*Добавление задания происходит аналогично.*

Выбирается тип фильтрации - *type*, *threads* - кол. потоков , *input* - входящий фаил, *output* - файл результатов.

*Второй шаг у разных модулей разный, а у модуля ping отсутствует:*

*regexp match*: выборка происходит по указанному вами регулярному выражению, два режима *if* - добавляет url если подстрока найдена, *unless* добавляет url если подстрока не найдена; одновременно можно сделать замену - *replace*, в первом поле регулярное выражение, во втором строка. В многопоточном режиме быстрее работать не будет, так что смысл ставить более одного потока - нет;

*check pr:* выборка по *google page rank*. Можно выбрать url'ы равные - *equal,* не равные - *unequal* , более - *more then* или менее - *less then* определенного значения *pr*. Модуль варьирует разные дц и юзер агенты, но google банит за сильно наглую проверку :) не так как при парсинге, но все же; *make csv* - делает файл csv формата *url,pr*; *check domain* - делает проверку не страницы, а домена этого сайта.

*scan urls:* выборка по регулярному выражению, аналогично *regexp match*, но скрипт смотрит не url а саму страницу этого url, т.е берет url, заходит по нему и сканирует на вхождение всю скаченную страницу;

*ping:* просто пингует ресурсы, добавляя только живые.

Последние два модуля особенно актуальны в многогопоточном режиме.

## VI. Настройки

*Основные настройки в разделе Profiles:*

*Use proxy:* использовать ли прокси.

*Connection timeout:* таймаут соединения, в секундах.

*Parsing tries:* количество попыток получить результат, при использовании прокси не менее 50 - 100, т.е если 100 раз подряд была плочая прокси, он переходит к следующему запросу.

*Thread delay:* задержка потока, в секундах, 0 - нет задержки, актуально без прокси.

*Reload proxy:* получать новые прокси каждые N минут.

*Unique entries:* удаляет повторы, по строке, т.е не будет одинаковых строк или по домену, т.е не будет строк с одинаковым доменом.

*Глобальные настройки доступны в Options:*

Устанавливается пароль - *password* на админку, *e-mail* для оповещения - *alert*.
*Where is python* - путь до интерпритатора питона, например /usr/bin/python, /usr/bin/local/python. *Where is zip* - путь до архиватора *zip.*
Так же доступны настройки модулей парсинга: маска запроса - mask и хост;
У модулей фильтрации индивидуальные настройки, во многом схожи с настройками профайлов парсинга; можно изменять некоторые системные файлы (справа), если вы не знаете для чего эти настройки - не изменяйте их).

## VII. Прокси

Сборщик прокси лучше поставить на крон. В разделе *Proxy* есть пример команды для вашего сервера - с интервалом 10 минут, если нужно ставьте чаще или реже.
Хотя запускать можно и вручную - *Manual start*.
Вы можете загрузить прокси напрямую - *list*. Сборщик собирает их с ресурсов (по одному на строку) - *sources*.
Прокси в источниках должны иметь формат *proxy:port*. Скрипт может пинговать прокси - *check proxy*, настраивается количество потоков - *threads* и таймаут.