LITL 0.3

URL shortener tool

Versions

0.1
original version

0.2
added error messages for API calls

0.3
log file formats
support for capcha and adverts in edit.cgi

Principle

LITL (shortened writing for “**li**t**tl**e”) is an URL shortener. That is, it transforms a long regular URL (such as, say, `https://www.google.com/search?source=hp&ei=t2jtWujUEY6RkwXb3apA&q=litl&oq=lit`) into a short URL (say `litl.be/AbCd1234`).

When somebody asks for the short URL (in this example `litl.be/AbCd1234`), everything happens just as if he typed in the long URL (that is, in this example, `https://www.google.com/search?source=hp& ei=t2jtWujUEY6RkwXb3apA&q=litl&oq=lit`).

There are many advantage to this process:

- The URL is obviously quicker and most often easier to type, especially for very long URLs.

- In some contexts where space is limitted (for instance tweets), a shorter URL is an advantage.

- Generation of bar codes or QR codes are greatly ameliorated if you use short URLs.

- It allows somebody to monitor the quantity of peoples following a link.

- It allows to hide a given URL destination: you know there is a link, but you don’t know this link leads to `google.com` (nor does search engines, indexers, etc…).

API

Basically there are two scenarii:

You can create anonymously a short URL (using for instance the demo page on litl.be). This works, but there are limitations to this method, though: you cannot get the URL logs nor can you delete or change your URL at a later point (if you could, that would mean that just anybody could spy or change the URL you have generated, which isn't a good idea). There is also a limitation on the quantity of URL you can create this way.

You can create an account, obtain credits (by buying them, or because another account donates them to you) and use those credits to create URLs. In this case, you are not limitted to a given quantity of URLs per day (only the quantity of credits you possess limits you) and you can have access to logs and edition of any of the URLs you have created.

Here is an overview of the various functions provided:

Creating a short URL from a long one
`http://litl.be/create.cgi`

Create an account
`http://litl.be/account.cgi`

Give credits from an account to another
`http://litl.be/give.cgi`

Get URL logs
`http://litl.be/logs.cgi`

Change or delete an URL
`http://litl.be/edit.cgi`

Resolve (ie convert short to long) an URL
`http://litl.be/url.cgi`

create.cgi

`litl.be/create.cgi?account=`*`accnt`***`&url=`***`long_url`*

`account` (optional)

the account ID, a long alphanumeric string which should be kept private as this is your access to litl functionalities.
If no account is provided, this is an anonymous link creation: you will not have access to link edition and logs for this link, and you will be able to create only 10 links per day for a given IP.

`url` (mandatory, optional if account provided)

The long URL you want to transform into a short one.
Please note that if you call two times `litl.be/create.cgi` on the same long URL, you will recieve two different results. This isn't a bug but a functionality: two persons may want to link to the same page, yet they certainly wouldn't want the logs, edition capabilities etc... to be shared. Hence the same long URL can (and should) have multiple shortened versions.

Examples:

Anonymous creation of an URL:

```
litl.be/create.cgi?url=http%3A%2F%2Fwww.google.com
```

will return

```
http://litl.be/AbCd0123
```
or
```
MSG: your quota has been exceeded, sorry.
MSG: bad url.
MSG: cannot shorten litl url.
```

Registered creation of an URL:

```
litl.be/create.cgi?account=0123456789&url=http%3A%2F%2Fwww.google.com
```

will return

```
http://litl.be/AbCd0123
```
or
```
MSG: your quota has been exceeded, sorry.
MSG: bad url.
MSG: bad key.
MSG: cannot shorten litl url.
```

Knowing your current account creation credit:

```
litl.be/create.cgi?account=0123456789
```

will return

```
MSG: your quota is 25 credits.
```

account.cgi

`litl.be/account.cgi?account=``accnt`

`account` (mandatory)

the account ID, a long alphanumeric string which should be kept private as this is your access to litl functionalities.
This is the account that asks for creation of another account (that is: anonymous users cannot create accounts). The account creation requires ten credits, that is the equivalent of 10 link creations.

Notes:

Note that the account have access to data/URLs created by its sub-accounts.

Examples:

Creation of an account:

```
litl.be/account.cgi?account=0123456789
```

will return

`9876543210` (that is, a new "sub-account" number)
or

```
MSG: your quota has been exceeded, sorry.
MSG: bad key.
```

Knowing your current account creation credit:

```
litl.be/create.cgi?account=0123456789
```

will return

```
MSG: your quota is 25 credits.
```

Since an account creation requires 10 credits, you can create two accounts.

give.cgi

`litl.be/give.cgi?account=`*`accnt`*`&to=`*`accnt2`*

`account` (mandatory)

the account ID, a long alphanumeric string which should be kept private as this is your access to litl functionalities.
This is the account that gives the credits.

`to` (mandatory)

another account ID, a long alphanumeric string.
This is the account that recieves the credits. Transfer costs no credits.

Note:

This functionality is extremely usefull in two circumstances: if you represent a firm and have multiple departements/projects, you can buy a stock of credits linked to a main account and then distribute them to various sub-accounts at your best convenience.

If you plan reselling litl.be services in any way, this is also the way to make it.

Examples:

Give 10 credits from account 123 to account 321:

```
litl.be/give.cgi?account=123&to=321&qtt=10
```

will return

```
OK
```
or
```
MSG: your quota has been exceeded, sorry.
MSG: bad source key
MSG: bad destination key
MSG: missing quantity
```

Getting back those 10 credits:

```
litl.be/give.cgi?account=321&to=123&qtt=10
or
litl.be/give.cgi?account=123&to=321&qtt=-10
```

will return

```
OK
```
or
```
MSG: your quota has been exceeded, sorry.
```

If needed, this is the perfect demonstration that an account ID should stay private !

Knowing your current account creation credit:

```
litl.be/create.cgi?account=123
```

will return

```
MSG: your quota is 25 credits.
```

edit.cgi

`litl.be/edit.cgi?account=`*`accnt`*`&short=`*`shorturl`*`&url=`*`longurl`*

`account` (mandatory)

the account ID, a long alphanumeric string which should be kept private as this is your access to litl functionalities.
This is the account to which belongs the short URL (or one of its parent accounts).

`short` (mandatory)

the URL that needs to be modified (either the full notation `http://litl.be/AbCd1234`, or just the code `AbCd1234`)

`url` (optional)

the new long URL associated with the short URL, if no long URL is provided, this disables the short URL.

`capcha` (optional, not implemented)

"`yes`" occasionally the visitor is prompted to enter a code
"`allways`" the visitor is prompted everytime
"`no`" the visitor is not prompted

`advert` (optional, not implemented)

"`no`" goes straight to the long URL configured
"`yes`" display an advertisement page with a link to the long URL

Note:

Since parent accounts have access to children account data, if you represent a firm/reseller, you can manage/process the logs or links for your departements/projects/customers.

Examples:

Disable the short link litl.be/AbCd1234:

```
litl.be/edit.cgi?account=123&short=AbCd1234
```

will return

```
OK
```

or

```
MSG: cannot access short link AbCd1234, sorry.
MSG: bad key
MSG: bad short
```

Change short link AbCd1234 to http://www.yahoo.com

```
litl.be/edit.cgi?account=321&short=AbCd1234&url=http%3A%2F%2Fwww.yahoo.c
om
```

will return

```
OK
```

or

```
MSG: cannot access short link AbCd1234, sorry.
MSG: bad key
MSG: bad short
```

url.cgi

`litl.be/url.cgi?short=`*`shorturl`*

`short` (mandatory)

the URL that needs to be converted to long version (either the full notation `http://litl.be/AbCd1234`, or just the code `AbCd1234`)

`ip` (optional)

IP address of the visitor

`referer` (optional)

referer of the visitor

Note:

Call to this .cgi will be logged as an access to the long URL.

If IP and/or referer are provided, they will be logged as so.

Characters that follows the short ID (for instance "/something" in `http://litl.be/AbCdEfGh/`**`something`**) are ignored but will be logged. This allows for some identification of the source, user or whatever other custom use you may think of.

Examples:

Convert litl.be/AbCd1234 to its long equivalent:

```
litl.be/url.cgi?short=AbCd1234
```

will return

```
https://www.google.com
```
or
```
MSG: not found
```

logs.cgi

`litl.be/logs.cgi?account=`*`accnt`***`&short=`***`shorturl`***`&format=`***`fmt`*

`account` (mandatory)

the account ID, a long alphanumeric string which should be kept private as this is your access to litl functionalities.
This is the account to which belongs the short URL (or one of its parent accounts).

`short` (mandatory)

the URL you want the logs of (either the full notation `http://litl.be/AbCd1234`, or just the code `AbCd1234`)

`format` (optional)

can be **`raw`**, **`csv`**, **`json`** or **`xml`**.
those formats are described further in the documentation.

Examples:

Get logs for litl.be/AbCd1234:

```
litl.be/edit.cgi?account=123&short=AbCd1234&format=csv
```

will return

```
"time";"short_id";"status";"ip";"referer";"request";"user_agent";"url"
…
```

Log file format (internal)

a serie of lines with the following information:

```
time|short_id|status|ip|referer|request|user_agent|url
```

where

time
time of request in YYYY/MM/DD hh:mm:ss format

short_id
short ID requested (that is abcdefgh in litl.be/abcdefgh)

status
"`ok`" if the short ID exists
"`not found`" if the short ID doesn't exist
"`disabled`" if the short ID exists but has been disabled

ip
IP address of the visitor

referer
referer page (where the short link is)

request
the full request (that is all that follows the litl.be). This is likely about the same than short_id, but you can decide to add some more information for your convenience. For instance you could use litl.be/abcdefgh/customer_id or litl.be/abcdefgh/page...

user_agent
user agent (that is browser version) used by visitor

url
the long url. Theoretically, this will allways be the same for a given short ID, but this may change if you decide to edit the URL at a later stage using edit.cgi

CSV format

Using excel somewhat non-standard convention of a semi-colon separator. The first line contains collumns titles, that is:

```
"time";"short_id";"status";"ip";"referer";"request";"use
r_agent";"url"
```

Followed by one line per request (using the same CSV format).

JSON format

```
[
   {
   “time”:”…”,
   “short_id”:”…”,
   “status”:”…”,
   “ip”:”…”,
   “referer”:”…”,
   “request”:”…”,
   “user_agent”:”…”,
   “url”:”…”
   },
   …
]
```

XML format

```
<requests>
   <request>
      <time>…</time>
      <short_id>…</short_id>
      <status>…</status>
      <ip>…</ip>
      <referer>…</referer>
      <request>…</request>
      <user_agent>…</user_agent>
      <url>…</url>
   </request>
   …
</requests>
```