A smul python tech talk

Creating a URL shurtening website

Author: João Barreiros (@unstablectrl)

Company: Runtime Revolution

What is a smul URL? (1)

```
https://www.google.com/search?
hl=en&ei=csxIYJX4CsHPgwfj5onYCQ&q=What+is+a+smul+URL%3
F&oq=What+is+a+smul+URL%3F&gs_lcp=Cgdnd3Mtd2l6EAMyBwgA
EEcQsAMyBwgAEEcQsAMyBwgAEEcQsAMyBwgAEEcQsAMyBwgAEEcQsA
MyBwgAEEcQsAMyBwgAEEcQsAMyBwgAEEcQsANQAFgAYNo7aAFwAXgA
gAFxiAFxkgEDMC4xmAEAqgEHZ3dzLXdpesgBCMABAQ&sclient=gws
-wiz&ved=0ahUKEwiV-eSM7aXvAhXB5-
AKHWNzApsQ4dUDCAw&uact=5
```

What is a smul URL? (2)

`https://www.google.com/...` → `https://smul.io/4579518`

Base 10 `[0-9]`

`10 ^ 7 = 10 Million`

Base 62 `[A-Z, a-z, 0-9]`

`62 ^ 7 = 3.5 Trillion`

Solutions?

Shurt id (1)

Solution 1 - Hashing the url

Hashing `https://www.google.com` would give us:

Function	Hash
Adler32	5f890849
CRC32	331e5b6b
MD5	8ffdefbdec956b595d257f0aaeefd623
SHA-1	ef7efc9839c3ee036f023e9635bc3b056d6ee2db

Shurt id (2)

Solution 2 - Counter

Counter `[0 ... ∞]` → Base 62 encode

`4579518` → `SMUL` → `https://smul.io/SMUL`

Encoding (1)

```
 123456789ABCDEFGH JKLMN PQRSTUVWXYZabcdefghijk mnopqrstuvwxyz
= 58 characters = base58

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
= 62 characters = base62

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz+/
= 64 characters = base64
```

Encoding (2)

Index	Binary	Char	Index	Binary	Char	Index	Binary	Char	Index	Binary	Char
0	000000	A	16	010000	Q	32	100000	g	48	110000	w
1	000001	B	17	010001	R	33	100001	h	49	110001	x
2	000010	C	18	010010	S	34	100010	i	50	110010	y
3	000011	D	19	010011	T	35	100011	j	51	110011	z
4	000100	E	20	010100	U	36	100100	k	52	110100	0
5	000101	F	21	010101	V	37	100101	l	53	110101	1
6	000110	G	22	010110	W	38	100110	m	54	110110	2
7	000111	H	23	010111	X	39	100111	n	55	110111	3
8	001000	I	24	011000	Y	40	101000	o	56	111000	4
9	001001	J	25	011001	Z	41	101001	p	57	111001	5
10	001010	K	26	011010	a	42	101010	q	58	111010	6
11	001011	L	27	011011	b	43	101011	r	59	111011	7
12	001100	M	28	011100	c	44	101100	s	60	111100	8
13	001101	N	29	011101	d	45	101101	t	61	111101	9
14	001110	O	30	011110	e	46	101110	u			
15	001111	P	31	011111	f	47	101111	v			

Encoding (3)

```
def encode(number):
    if number < 1: return CHARSET[0]
    code = ""
    while number > 0:
        number, remainder = divmod(number, BASE)
        code = CHARSET[remainder] + code
    return code
```

Value for A:

`A=0, AA=00, 9=61, BA=62 (Recommended for maths)`

Encoding (4)

```
def encode(number):
    code = ""
    while number > 0:
        number, remainder = divmod(number - 1, BASE)
        code = CHARSET[remainder] + code
    return code
```

Value for A:

```
A=1, AA=63, 9=62, BA=125 (Recommended for crypto)
```

Smul with Python

Packages

Flask

web development,
one drop at a time

Flask SQLAlchemy

Flask WTF

Project structure

```
smul/
├── static
│   └── style.css
├── templates
│   └── index.html
├── smul.py
└── requirements.txt
```

Database overview

Counter

Attribute	Type	Index	Unique	Nullable
value	int	false	false	false

Shurt

Attribute	Type	Index	Unique	Nullable
url	str	true	true	false
code	str	true	true	false

Let's do some coding

References

Url shortening system design

https://www.geeksforgeeks.org/system-design-url-shortening-service/

Base 62 encoding definition

https://en.wikipedia.org/wiki/Base62

Base 26 encoding explained

https://www.dcode.fr/base-26-cipher

Base N conversion algorithms

https://www.dcode.fr/base-n-convert/

That's all she wrote