

It's All Fake — Test Data Generation in D

Dr. Robert Schadek

DConf Online 2022

<https://github.com/symmetryinvestments/faked>

We have all been there 1/2

```
1 string randomForename() {  
2     return "Mike";  
3 }
```

We have all been there 1/2

```
1 string randomForename() {  
2     return "Mike";  
3 }
```

```
4 string randomForename2() {  
5     return choice(["Mike", "John", "Walter"]);  
6 }
```

We have all been there 1/2

```
1 string randomForename() {
2     return "Mike";
3 }

4 string randomForename2() {
5     return choice(["Mike", "John", "Walter"]);
6 }

7 string randomChineseNames() {
8     return choice(["婵娟", "冬梅"]);
9 }
```

We have all been there 2/2

```
15 string randomStreetName() {
16     bool t = uniform(0.0, 1.0) < 0.5;
17     string name = t
18         ? randomForename2()
19         : randomChineseNames();
20     return name ~ "'s Str.";
21 }
```

Surely, somebody has done this before

Enter `faker.js`

Enter faker.js

- Locations - Generate valid looking Addresses, Zip Codes, Street Names, States, and Countries!
- Time-based Data - Past, present, future, recent, soon... whenever!
- Localization - Set a locale to generate realistic looking Names, Addresses, and Phone Numbers.
- Finance - Create stubbed out Account Details, Transactions, and Crypto Addresses.
- Products - Generate Prices, Product Names, Adjectives, and Descriptions.
- Hacker Jargon - “Try to reboot the SQL bus, maybe it will bypass the virtual application!”
- Names - Generate virtual humans with a complete online and offline identity.
- Numbers - Of course, we can also generate random numbers and strings.

- en, en_AU, en_AU_ocker, en_BORK, en_CA, en_GB, en_GH, en_IE, en_IN, en_NG, en_US, en_ZA
- fr, fr_BE, fr_CA, fr_CH
- de, de_AT, de_CH
- es, es_MX
- nl, nl_BE
- pt_BR, pt_PT
- zh_CN, zh_TW
- af_ZA
- ar
- az
- cz
- dv
- ...

en/finance/account_type.js

```
1 module["exports"] = [  
2   "Checking",  
3   "Savings",  
4   "Money Market",  
5   'Investment',  
6   "Home Loan",  
7   "Credit Card",  
8   "Auto Loan",  
9   "Personal Loan"  
10  ];
```

```
1  class Faked {  
2    Random rnd;  
3  
4    string financeAccountType() {  
5      auto data = [  
6        "Checking",  
7        "Savings",  
8        "Money Market",  
9        "Investment",  
10       "Home Loan",  
11       "Credit Card",  
12       "Auto Loan",  
13       "Personal Loan"  
14     ];  
15     return choice(data, this.rnd);  
16   }  
17 }
```

en/cell_phone/formats.js

```
1 export default [
2     '###-###-####',
3     '(###) ###-####',
4     '1-###-###-####',
5     '###.###.####',
6 ];
1 string cellPhoneFormats() {
2     auto data = [
3         "###-###-####",
4         "(###) ###-####",
5         "1-###-###-####",
6         "###.###.####"
7     ];
8     return this.digitBuild(choice(data, this.rnd));
9 }
```

en/name/name.js

```
1 export default [
2   '{{name.prefix}} {{name.first_name}} {{name.last_name}}',
3   '{{name.first_name}} {{name.last_name}} {{name.suffix}}',
4   '{{name.first_name}} {{name.last_name}}',
5   '{{name.first_name}} {{name.last_name}}',
6   '{{name.first_name}} {{name.last_name}}',
7   '{{name.first_name}} {{name.last_name}}',
8 ];
9
1 string nameName() {
2   switch(uniform(0, 6, this.rnd)) {
3     case 0: return namePrefix() ~ " " ~ nameFirstName() ~ " " ~ nameLastName();
4     case 1: return nameFirstName() ~ " " ~ nameLastName() ~ " " ~ nameSuffix();
5     case 2: return nameFirstName() ~ " " ~ nameLastName();
6     case 3: return nameFirstName() ~ " " ~ nameLastName();
7     case 4: return nameMaleFirstName() ~ " " ~ nameLastName();
8     case 5: return nameFemaleFirstName() ~ " " ~ nameLastName();
9   }
10 }
```

File parsing 1/2

```
1 string[] prefixes =
2   [ "module[\"exports\" ] = "
3     , "module.exports = "
4     , "module['exports'] = "
5     , "export default ["
6     , "export default Object.freeze(["
7     , "export default {"
8   ];
```

```
1 string[] postfixes =
2   [ "];"
3     , "}"
4     , "};"
5     , "]"
6     , "]);"
7   ];
```

The rest is some boring `std.string` code and a lot of `std.format`

hand translation ... sometimes you have to

classes are awesome

```
1 class Base {
2     string name() {
3         return "John";
4     }
5
6     string streetName() {
7         return name() ~ "'s Str.";
8     }
9 }
```

```
11 class German : Base {
12     override string name() {
13         return "Manfred";
14     }
15 }
16
17 unittest {
18     Base b = new German();
19     assert(b.streetName() ==
20         "Manfred's Str.");
```


Resulting library is called faked

- 57 files
- ~ 220k lines
- 148 methods

Generator

- four files
- ~ 2.8k lines

Conclusion

- NPM has 1.2 million package
- there are no new ideas
- grab what you can and give nothing back

The End
