JSON::Validator

```perl
use JSON::Validator;
use Mojo::File;
use Mojo::JSON 'decode_json';

my $validator = JSON::Validator->new;
$validator->schema('address-schema.json');

my $data = Mojo::File->new('address.json')->slurp;
my $document = decode_json $data;

my @errors = $validator->validate($document);

die "@errors" if @errors;

say "lgtm";
```

```perl
my $validator = JSON::Validator->new;
$validator->schema('address-schema.json');
```

```perl
my $data = Mojo::File->new('address.json')->slurp;
my $document = decode_json $data;
my @errors = $validator->validate($document);
```

```
die "@errors" if @errors;

say "lgtm";
```

JSON::Validator

```perl
use JSON::Validator;
use Mojo::File;
use Mojo::JSON 'decode_json';

my $validator = JSON::Validator->new;
$validator->schema('address-schema.json');

my $data = Mojo::File->new('address.json')->slurp;
my $document = decode_json $data;

my @errors = $validator->validate($document);

die "@errors" if @errors;

say "lgtm";
```

JSON Schema

- enum—value must be one of the enumerated values
- oneOf, anyOf, allOf, not—array items
- references—pointers to other schemas
- regular expressions—not PCRE, but functional
- string length, array counts, property counts, integer maximum/minimum
- default values
- formats—date-time, email, hostname, ipv4, ipv6, uri, etc.