Plugins

```perl
#!/usr/bin/env perl
use Mojolicious::Lite;
use lib 'lib';

plugin 'TypeCast';

post '/' => sub {
    my $c = shift;
    my $json = $c->req->json;

    $c->to_integer($json, qw/foo bar/);

    $c->render(json => $json);
};

app->start;
```

```
package Mojolicious::Plugin::TypeCast;
use Mojo::Base 'Mojolicious::Plugin';

use Scalar::Util 'looks_like_number';

sub register {
    my ($self, $app, $conf) = @_;

    $app->helper(
        to_integer => sub {
            my ($self, $hash, @keys) = @_;

            for my $key (@keys) {
                next unless exists $hash->{$key};
                $hash->{$key} ||= 0;  ## empty, undef are ok
                next unless looks_like_number($hash->{$key});
                $hash->{$key} += 0;   ## convert to number
            }
        }
    );
}

1;
```

```
        to_integer => sub {
            my ($self, $hash, @keys) = @_;

            for my $key (@keys) {
                next unless exists $hash->{$key};
                $hash->{$key} ||= 0;  ## empty, undef are ok
                next unless looks_like_number($hash->{$key});
                $hash->{$key} += 0;   ## convert to number
            }
        }
```

```
plugin 'TypeCast';
```

```
$c->to_integer($json, qw/foo bar/);
```

Plugins

```perl
package Mojolicious::Plugin::TypeCast;
use Mojo::Base 'Mojolicious::Plugin';

use Scalar::Util 'looks_like_number';

sub register {
    my ($self, $app, $conf) = @_;

    $app->helper(
        to_integer => sub {
            my ($self, $hash, @keys) = @_;

            for my $key (@keys) {
                next unless exists $hash->{$key};
                $hash->{$key} ||= 0;  ## empty, undef are ok
                next unless looks_like_number($hash->{$key});
                $hash->{$key} += 0;   ## convert to number
            }
        }
    );
}

1;
```

```perl
#!/usr/bin/env perl
use Mojolicious::Lite;
use lib 'lib';

plugin 'TypeCast';

post '/' => sub {
    my $c = shift;
    my $json = $c->req->json;

    $c->to_integer($json, qw/foo bar/);

    $c->render(json => $json);
};

app->start;
```

Odds and Ends