とりあえず.

```
$ mojo generate lite-app
```

css や js など静的ファイルは public フォルダに置く.
プログラムでファイルを作成してダウンロードできるようにするには
そのファイルも public 以下に生成させると， hoge でアクセスできる.
外部テンプレートフォルダは templates で，hoge.html.ep などを置く.

■ クライアントとして

```
$ mojo get https://mojolicious.org
$ ./myapp.pl get /foo
$ ./myapp.pl get -M PUT -c '{"message":"Hello Mojo!"}' /hoge
```

サンプル

■ モデル

自動で生成されるのは，コントローラとビューの部分.
ある程度規模が大きいものは，シングルファイルでなく，モジュールを用意する.
つまり，コントローラ部分に埋め込まず，モデルを別ファイルに自前で書く.

```
# mkdir -p lib/MyApp/Model
# touch lib/MyApp/Model/Greet.pm
# chmod 644 lib/MyApp/Model/Greet.pm
```

lib/MyApp/Model/Greet.pm

```
package MyApp::Model::Greet;

use strict;
use warnings;
use experimental 'signatures';

sub new ($class) {bless {}, $class}

sub greeting ($self, $name) {
   return "Hello $name";
}

1;
```

myapp.pl

```
#!/usr/bin/env perl
use Mojolicious::Lite -signatures;

use lib 'lib';
use MyApp::Model::Greet;

helper greet => sub {state $greet = MyApp::Model::Greet->new};

get '/hello/:who' => sub ($c) {
   my $w = $c->stash('who');
   my $msg = $c->greet->greeting($w);
   $c->render(text => $msg);
};

app->start;
```

Try !

```
$ ./myapp.pl get /hello/taro
Hello taro
```

■ フォームで JSON をやり取りするサンプル

myapp.pl

```
#!/usr/bin/env perl
use Mojolicious::Lite -signatures;

get '/' => sub ($c) {
  $c->render(template => 'index');
};

get '/form' => sub ($c) {
  $c->render(template => 'form');
};

put '/endpoint' => sub ($c) {
  my $hash = $c->req->json;
  $c->render(json => {n => ++$hash->{num}});
};

app->start;
__DATA__

@@ index.html.ep
% layout 'default';
% title 'Welcome';
<h1>Welcome to the Mojolicious real-time web framework!</h1>

@@ layouts/default.html.ep
<!DOCTYPE html>
<html>
  <head><title><%= title %></title></head>
  <body><%= content %></body>
</html>
```

templates/form.html.ep

```
% layout 'default';
% title 'Form Test';

Input number.
<form id="form_x">
   <label for="int">N:</label>
   <input type="text" id="int" name="int" required>
   <button type="submit">Send</button>
</form>

<div id="result"></div>

<script>
document.getElementById("form_x").addEventListener("submit", function(event) {
       event.preventDefault();
       const i = document.getElementById("int").value;
       fetch("/post", {
           method: "POST",
           headers: {
               "Content-Type": "application/json",
           },
           body: JSON.stringify({num: i}),
       })
       .then(response => response.json())
       .then(data => {
           document.getElementById("result").innerHTML = "next number: " + data.n;
       })
       .catch(error => {
           document.getElementById("result").innerHTML = "error: " + error;
       });
});
```

```
</script>
```

サーバ デプロイ

■ Hypnotoad の FeeBSD 用 rc

/usr/local/etc/rc.d/mojolicious

```
#!/bin/sh
#
# PROVIDE: mojolicious
# REQUIRE: DAEMON
# KEYWORD: shutdown

. /etc/rc.subr

name=mojolicious
rcvar=mojolicious_enable

command="/usr/local/bin/hypnotoad"
myprg="/home/mojolicious/myapp.pl"

start_cmd="${command} ${myprg}"
stop_cmd="${command} -s ${myprg}"

load_rc_config ${name}

run_rc_command "$1"
```

■ Nginx リバースプロキシの最小限セットアップ

nginx.conf

```
upstream myapp {
 server 127.0.0.1:8080;
}
server {
  listen 80;
  server_name localhost;
  location / {
    proxy_pass http://myapp;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
  }
}
```