

From Code to Community: Sponsoring The Perl and Raku Conference 2025 [Learn more](#)

SRI / Mojolicious-9.38 / lib / Mojolicious / Command / Author / generate / app.pm

```

1  package Mojolicious::Command::Author::generate::app;
2  use Mojo::Base 'Mojolicious::Command';
3
4  use Mojo::Util qw(class_to_file class_to_path decamelize);
5
6  has description => 'Generate Mojolicious application directory structure';
7  has usage       => sub { shift->extract_usage };
8
9  sub run {
10     my ($self, $class) = (shift, shift || 'MyApp');
11
12     # Script
13     my $name = class_to_file $class;
14     $self->render_to_rel_file('mojo', "$name/script/$name", {class => $class});
15     $self->chmod_rel_file("$name/script/$name", 0744);
16
17     # Application class
18     my $app = class_to_path $class;
19     $self->render_to_rel_file('appclass', "$name/lib/$app", {class => $class});
20
21     # Config file (using the default moniker)
22     $self->render_to_rel_file('config', "$name/@{[decamelize $class]}.yml");
23
24     # Controller
25     my $controller = "${class}::Controller::Example";
26     my $path       = class_to_path $controller;
27     $self->render_to_rel_file('controller', "$name/lib/$path", {class => $controller});
28
29     # Test
30     $self->render_to_rel_file('test', "$name/t/basic.t", {class => $class});
31
32     # Static file
33     $self->render_to_rel_file('static', "$name/public/index.html");
34     $self->create_dir("$name/public/assets");
35
36     # Templates
37     $self->render_to_rel_file('layout', "$name/templates/layouts/default.html.ep");
38     $self->render_to_rel_file('welcome', "$name/templates/example/welcome.html.ep");
39 }
40
41 1;
-   Show 63 lines of Pod

```

```

105 __DATA__
106
107 @@ mojo
108 #!/usr/bin/env perl
109
110 use strict;
111 use warnings;
112
113 use Mojo::File qw(curfile);
114 use lib curfile->dirname->sibling('lib')->to_string;
115 use Mojolicious::Commands;
116
117 # Start command line interface for application
118 Mojolicious::Commands->start_app('<%= $class %>');
119
120 @@ appclass
121 package <%= $class %>;
122 use Mojo::Base 'Mojolicious', -signatures;

```

```

124 # This method will run once at server start
125 sub startup ($self) {
126
127
128     # Load configuration from config file
129     my $config = $self->plugin('NotYAMLConfig');
130
131     # Configure the application
132     $self->secrets($config->{secrets});
133
134     # Router
135     my $r = $self->routes;
136
137     # Normal route to controller
138     $r->get('/')->to('Example#welcome');
139 }
140
141 1;
142
143 @@ controller
144 package <%= $class %>;
145 use Mojo::Base 'Mojolicious::Controller', -signatures;
146
147 # This action will render a template
148 sub welcome ($self) {
149
150     # Render template "example/welcome.html.ep" with message
151     $self->render(msg => 'Welcome to the Mojolicious real-time web framework!');
152 }
153
154 1;
155
156 @@ static
157 <!DOCTYPE html>
158 <html>
159     <head>
160         <title>Welcome to the Mojolicious real-time web framework!</title>
161     </head>
162     <body>
163         <h2>Welcome to the Mojolicious real-time web framework!</h2>
164         This is the static document "public/index.html",
165         <a href="/">click here</a> to get back to the start.
166     </body>
167 </html>
168
169 @@ test
170 use Mojo::Base -strict;
171
172 use Test::More;
173 use Test::Mojo;
174
175 my $t = Test::Mojo->new('<%= $class %>');
176 $t->get_ok('/')->status_is(200)->content_like(qr/Mojolicious/i);
177
178 done_testing();
179
180 @@ layout
181 <!DOCTYPE html>
182 <html>
183     <head><title><%= title %></title></head>
184     <body><%= content %></body>
185 </html>

```

```
186 @@ welcome
187 %< layout: 'default';
188 %% title 'Welcome';
189 <h2><%= $msg %></h2>
190 <p>
191     This page was generated from the template "templates/example/welcome.html.ep"
192     and the layout "templates/layouts/default.html.ep",
193     <%= link_to 'click here' => url_for %> to reload the page or
194     <%= link_to 'here' => '/index.html' %> to move forward to a static page.
195 </p>
196
197 @@ config
198 % use Mojo::Util qw(sha1_sum steady_time);
199 ---
200 secrets:
201   - <%= sha1_sum $$ . steady_time . rand %>
202
```