Organizing your application

```
"You don't know how paralyzing that is, that stare of a blank canvas is"
- Vincent van Gogh
```

Sinatra is a blank canvas. It can transform into a single-file application, an embedded administration page, an API server, or a full-fledged hundred page website. Each use case is unique. This chapter will provide some advice on specific situations, but you will need to experiment to determine the best structure of your code. Don't be afraid to experiment.

A Single File Application

Obviously the file system structure is easy for this one.

A single file can contain an entire multi-page application. Don't be afraid to use inline templates, multiple classes (this isn't Java, multiple classes will live happily next to each other in the same file)

A Large Site

This one is trickier. My advice is to look to Rails for advice. They have a well structured set of directories to hold many of the components that make up a larger application. Remember that this file structure is just a suggestion.

```
|- config.ru            # A rackup file. Load server.rb, and
|- server.rb            # Loads all files, is the base class
|- app/
\--- routes/
\------ users.rb
\--- models/
\------ user.rb         # Model. Database or not
\--- views/
\------ users/
\--------- index.erb
\--------- new.erb
\--------- show.erb
```

In `server.rb` it should be a barebones application.

```
class Server < Sinatra::Base
  configure do
    # Load up database and such
  end
end
```

```
# Load all route files
Dir[File.dirname(__FILE___) + "/app/routes/**"].each do |route|
  require route
end
```

And the route files look something like:

```
# users.rb
class Server < Sinatra::Base
  get '/users/:id' do
    erb :"users/show"
  end

  # more routes...
end
```