Fork me on GitHub

Sinatra Recipes

Community contributed recipes and techniques

Select a topic

Helpers ¶ ↑

Helpers are a great way to provide reusable functionality to your application. The helpers in this section have been contributed from members of the Sinatra community, to find out more about helpers check the Sinatra README.

- Partials
- Partials Using The Sinatra-partial Gem

Did we miss something?

It's very possible we've left something out, that's why we need your help!This is a community driven project after all. Feel free to fork the project and send us a pull request to get your recipe or tutorial included in the book.

See the README for more details.

Top

Sinatra Recipes

Community contributed recipes and techniques

Chapters

Implementation of Rails style partials

Using partials in your views is a great way to keep them clean. Since Sinatra takes the hands off approach to framework design, you'll have to implement a partial handler yourself.

Here is a really basic version:

```
# Usage: partial :foo
helpers do
  def partial(page, options={})
    haml page, options.merge!(:layout => false)
  end
end
```

A more advanced version that would handle passing local options, and looping over a hash would look like:

```
# Render the page once:
# Usage: partial :foo
#
# foo will be rendered once for each element in the array, passing in a local
# variable named "foo"
# Usage: partial :foo, :collection => @my_foos

helpers do
  def partial(template, *args)
    options = args.extract_options!
    options.merge!(:layout => false)
    if collection = options.delete(:collection) then
      collection.inject([]) do |buffer, member|
        buffer << haml(template, options.merge(
                                    :layout => false,
                                    :locals => {template.to_sym => member}
                                  )
                           )
      end.join("\n")
    else
      haml(template, options)
    end
  end
end
```

Did we miss something?

It's very possible we've left something out, that's why we need your help!This is a community driven project after all. Feel free to fork the project and send us a pull request to get your recipe or tutorial included in the book.

See the README for more details.

Top

Sinatra Recipes

Community contributed recipes and techniques

Fork me on GitHub

Select a topic

Chapters

Partials using the sinatra-partial ¶gem

An alternative to define helpers for using partials is to use the sinatra-partial gem.

Add the gem to your `Gemfile`:

```
gem 'sinatra-partial', require: 'sinatra/partial'
```

Register the extension (you don’t have to do this for classic style apps):

```
class SomeApp < Sinatra::Base
  configure do
    register Sinatra::Partial
  end

  # ...
end
```

And enjoy!

```
partial 'some_partial', template_engine: :erb
```

To avoid telling `partial` which template engine to use, you can configure it globally:

```
class SomeApp < Sinatra::Base
  configure do
    register Sinatra::Partial
    set :partial_template_engine, :erb
    end

    # ...
  end
```

More information¶ ↑

For more configuration options and usage, view the sinatra-partial gem on GitHub.

Did we miss something?

It's very possible we've left something out, that's why we need your help!This is a community driven project after all. Feel free to fork the project and send us a pull request to get your recipe or tutorial included in the book.

See the README for more details.

Top