

Error Handling

Error handlers run within the same context as routes and before filters, which means you get all the goodies it has to offer, like `haml`, `erb`, `halt`, etc.

Not Found

When a `Sinatra::NotFound` exception is raised, or the response's status code is 404, the `not_found` handler is invoked:

```
not_found do
  'This is nowhere to be found.'
end
```

Error

The `error` handler is invoked any time an exception is raised from a route block or a filter. But note in development it will only run if you set the show exceptions option to `:after_handler`:

```
set :show_exceptions, :after_handler
```

The exception object can be obtained from the `sinatra.error` Rack variable:

```
error do
  'Sorry there was a nasty error - ' + env['sinatra.error'].message
end
```

Custom errors:

```
error MyCustomError do
  'So what happened was...' + env['sinatra.error'].message
end
```

Then, if this happens:

```
get '/' do
  raise MyCustomError, 'something bad'
end
```

You get this:

```
So what happened was... something bad
```

Alternatively, you can install an error handler for a status code:

```
error 403 do
  'Access forbidden'
end

get '/secret' do
  403
end
```

Or a range:

```
error 400..510 do
  'Boom'
end
```

Sinatra installs special `not_found` and `error` handlers when running under the development environment to display nice stack traces and additional debugging information in your browser.