Accessing the Request Object

The incoming request object can be accessed from request level (filter, routes, error handlers) through the request method:

```
# app running on http://example.com/example
get '/foo' do
  t = %w[text/css text/html application/javascript]
  request.accept              # ['text/html', '*/*']
  request.accept? 'text/xml'  # true
  request.preferred_type(t)   # 'text/html'
  request.body                # request body sent by the client (see below)
  request.scheme              # "http"
  request.script_name         # "/example"
  request.path_info           # "/foo"
  request.port                # 80
  request.request_method      # "GET"
  request.query_string        # ""
  request.content_length      # length of request.body
  request.media_type          # media type of request.body
  request.host                # "example.com"
  request.get?                # true (similar methods for other verbs)
  request.form_data?          # false
  request["some_param"]       # value of some_param parameter. [] is a shortcut to
the params hash.
  request.referrer            # the referrer of the client or '/'
  request.user_agent          # user agent (used by :agent condition)
  request.cookies             # hash of browser cookies
  request.xhr?                # is this an ajax request?
  request.url                 # "http://example.com/example/foo"
  request.path                # "/example/foo"
  request.ip                  # client IP address
  request.secure?             # false (would be true over ssl)
  request.forwarded?          # true (if running behind a reverse proxy)
  request.env                 # raw env hash handed in by Rack
end
```

Some options, like script_name or path_info, can also be written:

```
before { request.path_info = "/" }

get "/" do
  "all requests end up here"
end
```

The request.body is an IO or StringIO object:

```
post "/api" do
  request.body.rewind  # in case someone already read it
  data = JSON.parse request.body.read
  "Hello #{data['name']}!"
end
```