Rack (web server interface)

Rack provides a modular and adaptable interface for developing web applications in Ruby. By wrapping HTTP requests and responses it unifies the API for web servers, web frameworks, and software in between (called middleware) into a single method call.

Rack is used by many Ruby web frameworks and libraries, such as Ruby On Rails and Sinatra. It is available as a Ruby Gem.

Rack has already inspired a JavaScript framework[1] (jackjs) and a Perl one (Plack), a Common Lisp one[2] (Clack), and has resulted in the Ruby developer quasi-standard of "rack-compliant".[3]

It was also cited as an inspiration for OWIN.[4]

1 Overview

The characteristics of a Rack application is that the application object responds to the call method. The call method takes in the Environment object as argument and returns the Rack response object.

1.1 Environment[5]

The environment that is taken as argument by the call method refers to an object that has:
a) Information on the HTTP Request This includes the information like:

- HTTP request method
- The URL information(information that would direct to the application, information that directs to the actual location in the application, Query string)
- Server information like the server name and server port
- The HTTP meta variables that are received from the client

b) Rack specific information This includes the information like

- The version of the Rack application that is running
- The URL scheme that is used, that is, if the request that is received is http or https.
- The raw HTTP data.
- A Ruby object for reporting errors.
- Information like if the application object is simultaneously invoked from another thread or process.
- Information on the server expectations and capabilities (capability of the server for connection hijacking).

In case the application is being used as a middleware, the environment can have objects that would provide session information, logging capabilities, information on the size of the data that can be used for read and writes etc. In addition to these, the server can store their own data in the environment.

1.2 Rack response[5]

The rack server object returns a response which contains three parts: the status, headers and the body.

- The status contains the HTTP status codes such as 200,404.
- The header contains the response for each and gives the key-value pairs. The keys have to be strings.
- Body contains the final data which is sent by the server to the requester.

Rack::Response provides a convenient interface to create a Rack response. The class Rack::Response is defined in lib/rack/response.rb. To use the Response class, instantiate it from the middleware layer down the stack. It can be used to modify the cookies.

1.3 Middleware in racks[5]

Rack makes it easy to add a chain of middleware components between the application and the web server. Multiple middleware components can be used in the rack which modifies the request/response before handing it over to the next component. This is called middleware stack.

The Rack server adds multiple middle middleware by default for the functionalities like showing exception with all the details,[6] validating the request and responses according to the Rack spec[7] etc.

2 Example application

A Rack-compatible "Hello World" application in Ruby syntax:

helloWorld.ru # The application that has the call method defined. class HelloWorld # Call method that would return the HTTP status code, the content type and the content. def call (env) [200, {"Content-Type" => "text/html; charset=utf-8"}, ["Hello World"]] end end

The server for the above code can be initiated using "rackup helloWorld.ru" and can be accessed at http://localhost:9292/ The default port used by the Rack application is 9292.

3 See also

- Python WSGI
- Perl PSGI
- Javascript JSGI
- Python Paste
- Smalltalk Seaside
- FastCGI
- Java Servlet
- Server-side JavaScript
- Apache JServ Protocol
- ZeroC Ice
- Cisco Etch
- ISAPI Internet Server Application Programming Interface (Microsoft)

4 References

[1] jack - introduction. Jackjs.org. Retrieved on 2013-09-20.

[2] clacklisp.org. Retrieved on 2014-10-17.

[3] Pancake: How To Stack and Loosely Couple Rack-Based Webapps Together. Rubyinside.com (2009-12-04). Retrieved on 2013-09-20.

[4] http://www.asp.net/aspnet/overview/owin-and-katana/an-overview-of-project-katana. Asp.net. Retrieved on 2014-10-01.

[5] "Documentation for rack". *www.rubydoc.info*. Retrieved 2016-09-14.

[6] "Rack::ShowExceptions". *www.rubydoc.info*. Retrieved 2016-09-14.

[7] "Rack::Lint". *www.rubydoc.info*. Retrieved 2016-09-14.

5 External links

- Official website

6 Text and image sources, contributors, and licenses

6.1 Text

- **Rack (web server interface)** *Source:* https://en.wikipedia.org/wiki/Rack_(web_server_interface)?oldid=770274731 *Contributors:* Daniel Brockman, Uzume, Rfl, LuckyStarr, Stesmo, FrenchIsAwesome, Pi Delport, Redconfetti, Story Weaver, SmackBot, Kokot.kokotisko, Jamie Lokier, Lindes, Tedickey, CommonsDelinker, Bhanks, Grshiplett, Diego.viola, Ken g6, Dvyjones, Jackfork, Lys1123, Sphilbrick, Asmega, Matthieu fr, Addbot, Mortense, MagnusA.Bot, MrOllie, Luckas-bot, Yobot, AnomieBOT, Omnipaedista, FrescoBot, Sae1962, Nyoungman, JnRouvignac, Karl Brodowsky, BG19bot, Julioalucero, Compfreak7, Honglilai, The Illusive Man, Myconix, EuAndreh, Joejoebob, Rkadeku, Asmotiwa, Learyseaward and Anonymous: 27

6.2 Images

- **File:Commons-logo.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/4a/Commons-logo.svg *License:* PD *Contributors:* ? *Original artist:* ?
- **File:Folder_Hexagonal_Icon.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/48/Folder_Hexagonal_Icon.svg *License:* Cc-by-sa-3.0 *Contributors:* ? *Original artist:* ?
- **File:Free_and_open-source_software_logo_(2009).svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/31/Free_and_open-source_software_logo_%282009%29.svg *License:* Public domain *Contributors:* FOSS Logo.svg *Original artist:* Free Software Portal Logo.svg (FOSS Logo.svg): ViperSnake151
- **File:Ruby_logo_64x64.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/f1/Ruby_logo_64x64.png *License:* CC BY-SA 2.5 *Contributors:* http://rubyidentity.org/ *Original artist:* Yukihiro Matsumoto

6.3 Content license

- Creative Commons Attribution-Share Alike 3.0