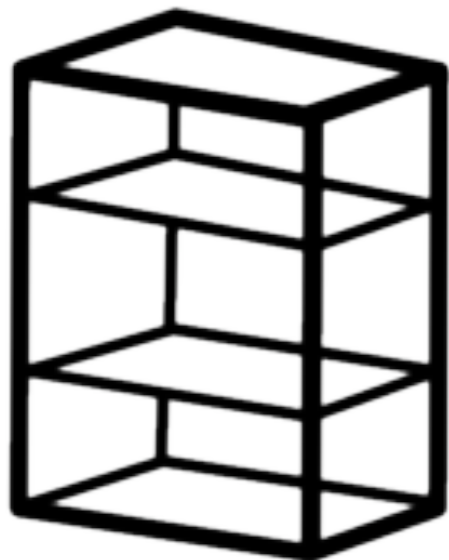


# Was ist Rack?

Christian Neukirchen

Lead Developer, Rack  
Editor in Chief, Trivium

<http://chneukirchen.org/>



**rack**  
powers web applications

# Vorstellung

- Rubyist seit 2001
- Autor von
  - *rubypants* (smart quotes in HTML)
  - *test/spec* und *bacon* (BDD)
  - *Rack*
- Im *wahren Leben* Mathematikstudent
- Ruby als Allroundsprache

# Was ist Rack?

- Eine Spezifikation
  - ein abstraktes Interface um Webserver und Webanwendungen zu verbinden
- Eine Referenzimplementierung derer
  - ...mit *vielen* nützlichen Utilities

# Hello, Rack!

```
class HelloRack
  def call(env)          # CGI-style environment
    [200,               # status
     {"Content-Type" => "text/html"}, # headers
     ["Hello, Rack!"]]  # body
  end
end
```

```
Rack::Handler::Mongrel.run HelloRack.new,
  :Port => 9999
```

# Vorteile von Rack (I)

- *Middleware*, Anwendungen die wiederum andere Anwendungen aufrufen
  - Authentifizierung (HTTP, OpenID)
  - Sessions
  - Routing
  - Caching
  - Logging
  - Debugging

# Vorteile von Rack (II)

- *Deployment*, da jeder Webserver die App aufrufen kann
  - Mongrel, Thin, Ebb, WEBrick...
  - Phusion Passenger (mod\_rack)
  - Heroku (lass' andere deployen)

# Vorteile von Rack (III)

- Weniger Arbeit
  - für Server-Entwickler
  - für Framework-Entwickler
  - für Anwendungs-Entwickler

# Wer verwendet Rack?

- *De-facto* Standard

- Frameworks

- Camping
- Halcyon
- Mack
- Maveric
- Merb
- **Rails**
- Ramaze
- Rum
- Sinatra
- Vintage
- Waves

- Webserver

- CGI
- Ebb
- **FastCGI**
- Fuzed
- JRuby Servlets
- LiteSpeed
- **Mongrel**
- **Phusion Passenger**
- SCGI
- Swiftcore
- Thin
- Unicorn
- WEBrick



# Wie profitiere *ich*?

- Als Rails-User
  - Middleware!
    - z.b. Rack::Cache, Rack::Test
  - *Rails Metal*
    - performance-kritische Teile ohne den Rails-Overhead schreiben
      - APIs, polling AJAX, ...

# Rails Metal

```
# config/environment.rb  
config.middleware.use(Rack::Cache, :verbose => true)
```

```
% script/generate metal foo  
% $EDITOR app/metal/foo.rb
```

```
class Foo  
  def self.call(env)  
    if env["PATH_INFO"] =~ /^\/foo/  
      ... handle the request ...  
    else  
      [404, {}, "Rails will handle this"]  
    end  
  end  
end  
end
```

# Ausblick

- Rack auch in andere Sprachen
  - Lua: WSAPI
  - Server-Side JavaScript: Jack
  - Perl 6: web.pm
  - Python: WSGI 2(!)

# Die Community

- Mailing-Liste: [rack-devel@googlegroups.com](mailto:rack-devel@googlegroups.com)
- IRC: #rack @ freenode.net
- Github: <http://github.com/rack>
  - dort auch *rack-contrib*, ein offenes Repository für Middleware
- Das Rack-Core-Team pflegt das Projekt (chneukirchen, raggi, josh, manveru, rtomayko, scytrin)

# Bis bald in Vegas?

## **The Future of Deployment**

05/05/2009 1:50pm – 2:40pm

Ballroom A, RailsConf 2009

*Danke für eure Aufmerksamkeit.*