



Grundlagen der Web-Entwicklung

INF3172

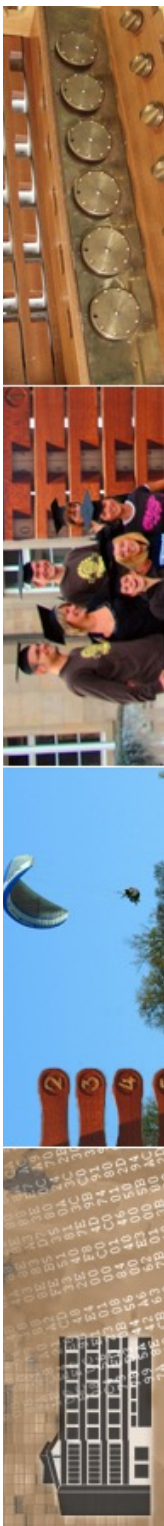
MVC-Frameworks:

Ruby on Rails & CakePHP

Thomas Walter

08.01.2026

Version 1.0

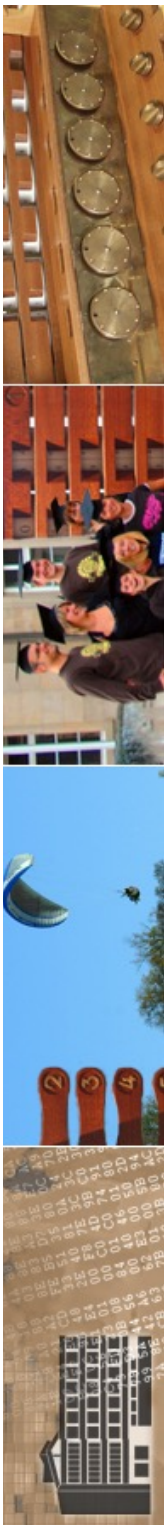


Klausur

- unsere Klausur findet statt:

Freitag, 06.02.2026, 12h - 14h, N10

- Nachklausur 27.03.2026, 12h - 14h, F119

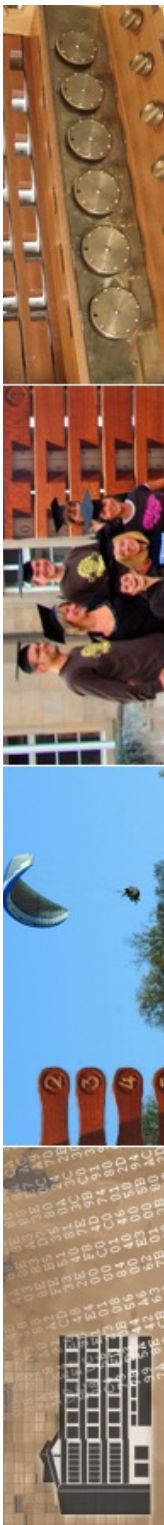


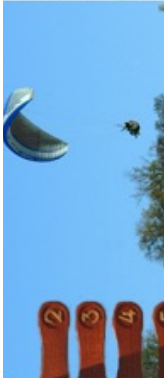
wir suchen: HiWis

- am ZDV sind häufig HiWi-Stellen zu besetzen
- aktuell suchen wir ab SoSe 2026:

HiWis für die Genius-Bar

- ein sehr spannender, abwechslungsreicher und lehrreicher (!) Job
- bei Interesse Mail mit dem Subject HiWi an thomas.walter@uni-tuebingen.de





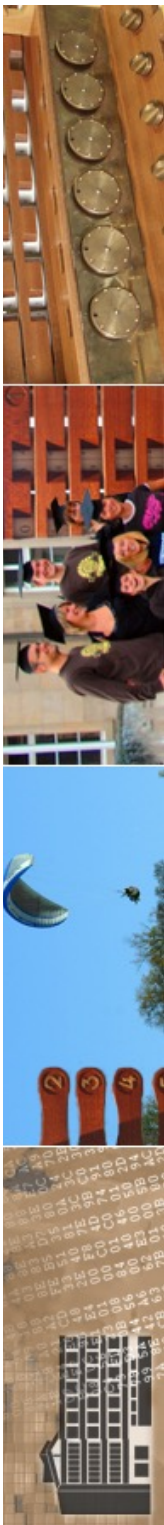


die Scriptsprache Ruby

- ein moderner Nachfolger von Perl
- sauber objektorientiert
- www.ruby-lang.org



Ruby
A Programmer's Best Friend





Install Docs Libraries Contribution Community News

English (en) ▾



A Programmer's Best Friend

Since 1995

Ruby

Latest Version: **4.0.0**

DOWNLOAD

```
# Output "I love Ruby"
say = "I love Ruby"
puts say
```

```
# Output "I *LOVE* RUBY"
say = say.sub("love", "*love*")
puts say.upcase
```

```
# Output "I *love* Ruby"
# five times
5.times { puts say }
```

```
# Ruby knows what you
# mean, even if you
# want to do math on
# an entire Array
cities = %w[ London
             Oslo
             Paris
             Amsterdam
             Berlin ]
visited = %w[Berlin Oslo]

puts "I still need " +
      "to visit the " +
      "following cities:",
      cities - visited
```

```
# The Greeter class
class Greeter
  def initialize(name)
    @name = name.capitalize
  end

  def salute
    puts "Hello #{@name}!"
  end
end

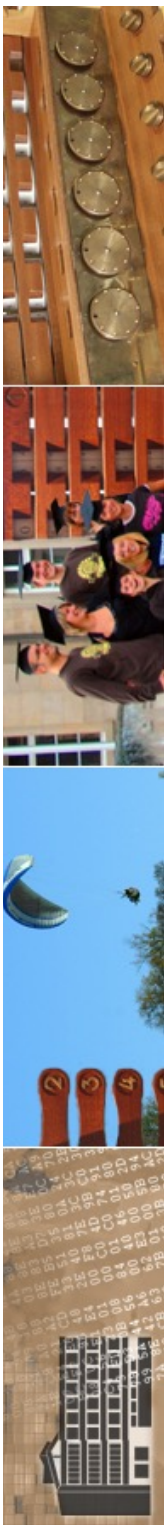
# Create a new object
g = Greeter.new("world")

# Output "Hello World!"
```



ruby

- entstanden durch
Yukihiro Matsumoto
(„Mats“)
 - aktuell 4.0.0
 - 3.4.8 und 2.7.7 stable
 - „Klassiker“: Versionen 1.9.x
 - moderne Sprache
 - OO





- **Stable releases:**

- [Ruby 4.0.0](#)

sha256: 2e8389c8c072cb658c93a1372732d9eac84082c88b065750db1e52a5ac630271

- [Ruby 3.4.8](#)

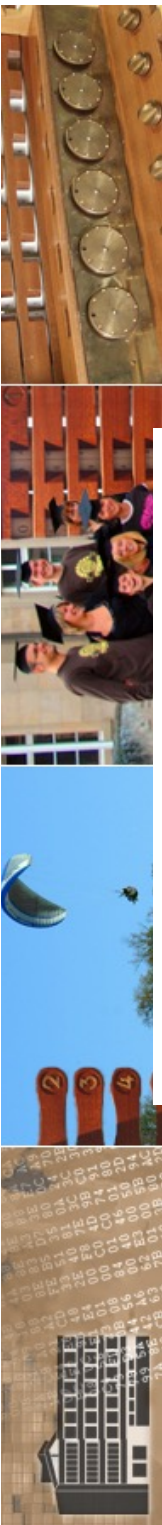
sha256: 53c4ddad41fbb6189f1f5ee0db57a51d54bd1f87f8755b3d68604156a35b045b

- [Ruby 3.3.10](#)

sha256: b555baa467a306cfc8e6c6ed24d0d27b27e9a1bed1d91d95509859eac6b0e928

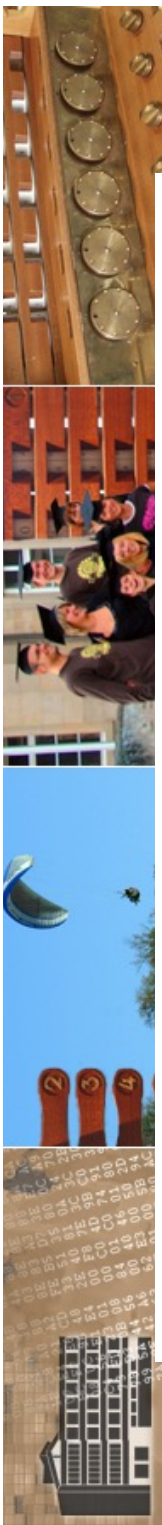
- [Ruby 3.2.9](#)

sha256: abbad98db9aeb152773b0d35868e50003b8c467f3d06152577c4dfed9d88ed2a



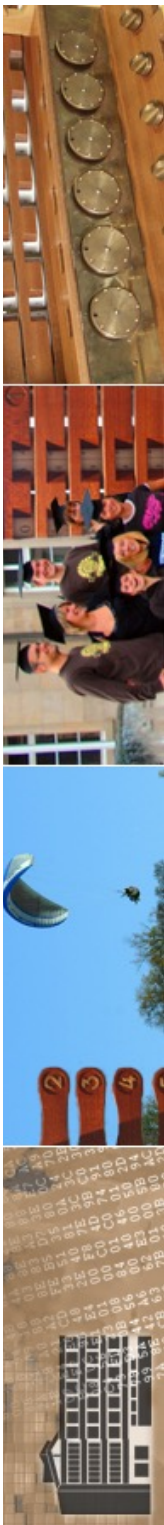


Release Version	Release Date	Download URL	Release Notes
Ruby 4.0.0	2025-12-25	download	more...
Ruby 4.0.0-preview3	2025-12-18	download	more...
Ruby 3.4.8	2025-12-17	download	more...
Ruby 4.0.0-preview2	2025-11-17	download	more...
Ruby 3.3.10	2025-10-23	download	more...
Ruby 3.4.7	2025-10-07	download	more...
Ruby 3.4.6	2025-09-16	download	more...
Ruby 3.3.9	2025-07-24	download	more...
Ruby 3.2.9	2025-07-24	download	more...
Ruby 3.4.5	2025-07-15	download	more...
Ruby 3.4.4	2025-05-14	download	more...
Ruby 3.5.0-preview1	2025-04-18	download	more...
Ruby 3.4.3	2025-04-14	download	more...
Ruby 3.3.8	2025-04-09	download	more...
Ruby 3.2.8	2025-03-26	download	more...
Ruby 3.1.7	2025-03-26	download	more...
Ruby 3.4.2	2025-02-14	download	more...
Ruby 3.2.7	2025-02-04	download	more...
Ruby 3.2.7	2025-02-15	download	more...



die Prinzipien von Rails

- Ruby on Rails (Rails)
 - Juli 2004: David Heinemeier Hansson
 - Implementiert MVC-Paradigma
 - fünf Module:
 - M: Active Record
 - V: Action Pack (Request- / Reponse-Behandlung)
 - Action-Mailer (Email)
 - Action Web-Service
 - Active Support (Ruby-Erweiterungen)





Frameworks (serverseitig) und MVC

	V	C	M
Smarty	X		
Laminas (ZF)	X	X	
R o R	X	X	X
Cake PHP	X	X	X



All Rails World 2025 talks now on YouTube.

Source

Docs

Community

News



Events

Jobs

Merch

Foundation

Compress the complexity of modern web apps.

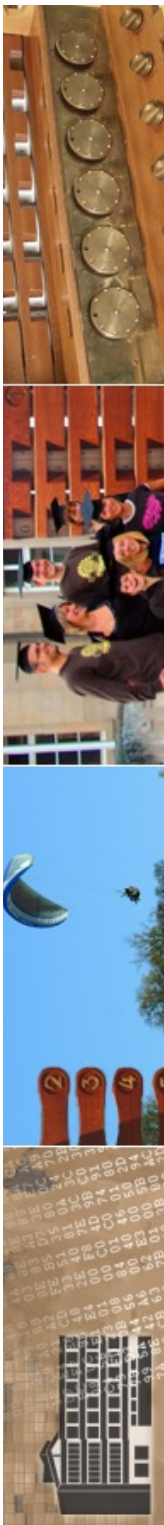
Learn just what you need to get started, then keep leveling up
as you go. **Ruby on Rails scales from HELLO WORLD to IPO.**

Rails 8.1.1 — released October 28, 2025



Prinzip

- „Konvention ersetzt Konfiguration“
 - anstelle aufwendiger Configs feste Konventionen, die Zusammenhänge definieren
 - Model-Klasse **Book** (Singular) bildet Tabelle **books** (Plural) ab
 - integriertes Mapping auf Controllermethoden





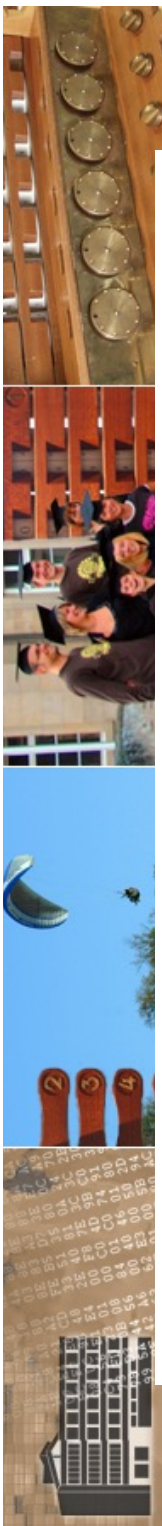
Convention over Configuration

One of the early productivity mottos of Rails went: “You’re not a beautiful and unique snowflake”. It postulated that by giving up vain individuality, you can leapfrog the toils of mundane decisions, and make faster progress in areas that really matter.

Who cares what format your database primary keys are described by? Does it really matter whether it’s “id”, “postId”, “posts_id”, or “pid”? Is this a decision that’s worthy of recurrent deliberation? No.

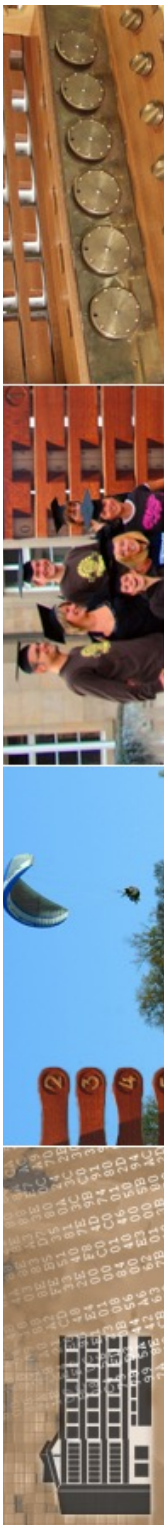
Part of the Rails’ mission is to swing its machete at the thick, and ever growing, jungle of recurring decisions that face developers creating information systems for the web. There are thousands of such decisions that just need to be made once, and if someone else can do it for you, all the better.

Not only does the transfer of configuration to convention free us from deliberation, it also provides a lush field to grow deeper abstractions. If we can depend on a Person class mapping to people table, we can use that same inflection to map an association declared as `has_many :people` to look for a Person class. The power of good conventions is that they pay dividends across a wide spectrum of use.



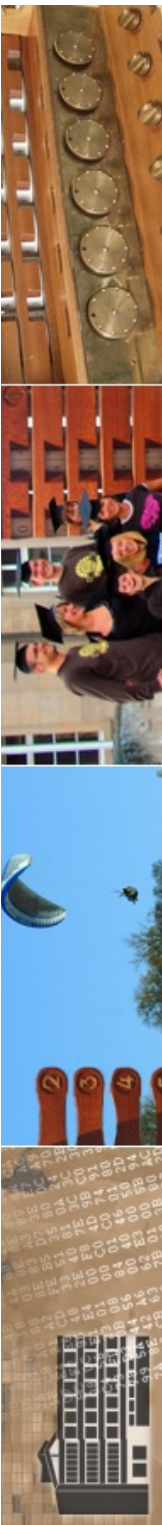
Scaffolding

- **Scaffolding** (Bauprinzip) erlaubt die Entwicklung von Web-Applikationen „On the fly“
 - Rails ideal für „Fast Prototyping“
- **CRUD:** Create, Read, Update, Delete aus Datenbank wird die Anwendung „on the fly“ Erzeugt



Webserverintegration

- Rails kommt mit eigenem Webserver (in Ruby geschrieben): WEBrick
 - zahlreiche andere auch möglich
- Produktionseinsatz:
 - Apache
 - mod_fastcgi sehr empfohlen
 - und möglichst noch mod_ruby



To begin with, let's get some text up on screen quickly. To do this, you need to get your Rails application server running.

4.1 Starting up the Web Server

You actually have a functional Rails application already. To see it, you need to start a web server on your development machine. You can do this by running the following command in the `blog` directory:

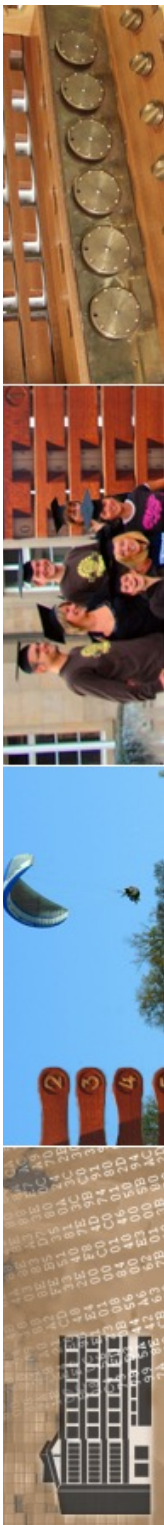


```
$ bin/rails server
```

Copy

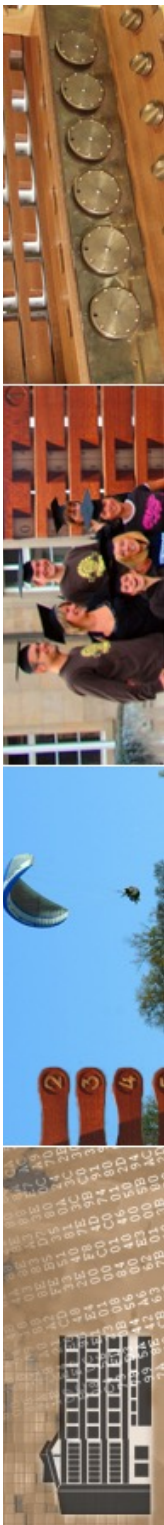
unterstützte DBMS

- momentan unterstützt Rails:
 - MySQL/MariaDB
 - PostgreSQL
 - DB2
 - Oracle
 - MS SQL Server
- wird Modul Active Record verwendet, muss eines dieser DBMS verwendet werden



M V C

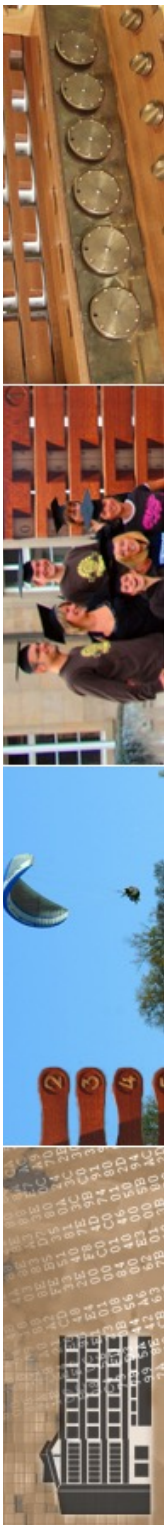
- Model: eines der unterstützten DBMS
- Controller: Instanz der ActionController-Klasse
- View: Klasse ActionView
 - HTML/XML/JavaScript/Binärdaten
 - Templates RHTML/RXML/RJS

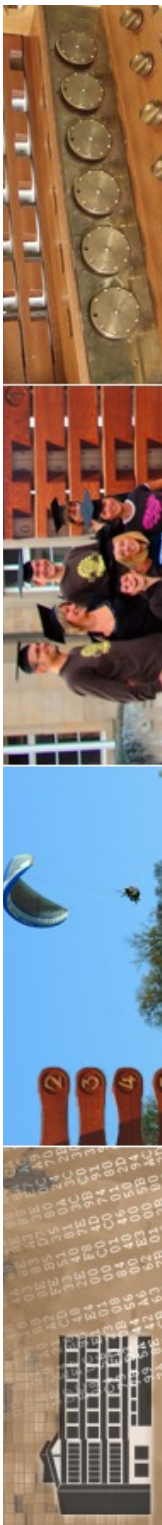




Ressourcen und Dokumentation

- Hauptressource:
 - www.rubyonrails.org
 - www.rubyonrails.de
- Dokumentation:
 - <http://api.rubyonrails.org>





app/models/article.rb

```
class Article < ApplicationRecord
  belongs_to :author, default: -> { Current.user }
  has_many :comments

  has_one_attached :cover_image
  has_rich_text :content, encrypted: true
  enum status: %i[ drafted published ]

  scope :recent, -> { order(created_at: :desc).limit(25) }

  after_save_commit :deliver_later, if: :published?

  def byline
    "Written by #{author.name} on #{created_at.to_s(:short)}"
  end

  def deliver_later
    Article::DeliveryJob.perform_later(self)
  end
end
```

Active Records make modeling easy.

Databases come to life with business logic encapsulated in rich objects. Modeling associations between tables, providing callbacks when saved, encrypting sensitive data seamlessly, and expressing SQL queries beautifully.

app/views/articles/show.html.erb

```
<h1><%= @article.title %></h1>

<%= image_tag @article.cover_image.url %>

<p><%= @article.content %></p>

<%= link_to "Edit", edit_article_path(@article) if Current.user.admin? %>
```

Action Views mix Ruby and HTML.

Templates can use the full versatility of Ruby, excessive code is extracted into helpers, and the domain model is used directly and interwoven with

Bildschirmfoto

app/controllers/articles_controller.rb

```
class ArticlesController < ApplicationController
  def index
    @articles = Article.recent
  end

  def show
    @article = Article.find(params[:id])
    fresh_when etag: @article
  end

  def create
    article = Article.create!(article_params)
    redirect_to article
  end

  private
  def article_params
    params.require(:article).permit(:title, :content)
  end
end
```

Action Controllers handle all requests.

Controllers expose the domain model to the web, process incoming parameters, set caching headers, and render templates, responding with either HTML or JSON.

config/routes.rb

```
Rails.application.routes.draw do
  resources :articles do
    # /articles, /articles/1
    resources :comments # /articles/1/comments, /comments/1
  end

  root to: "articles#index" # /
end
```

Action Dispatch route URLs.

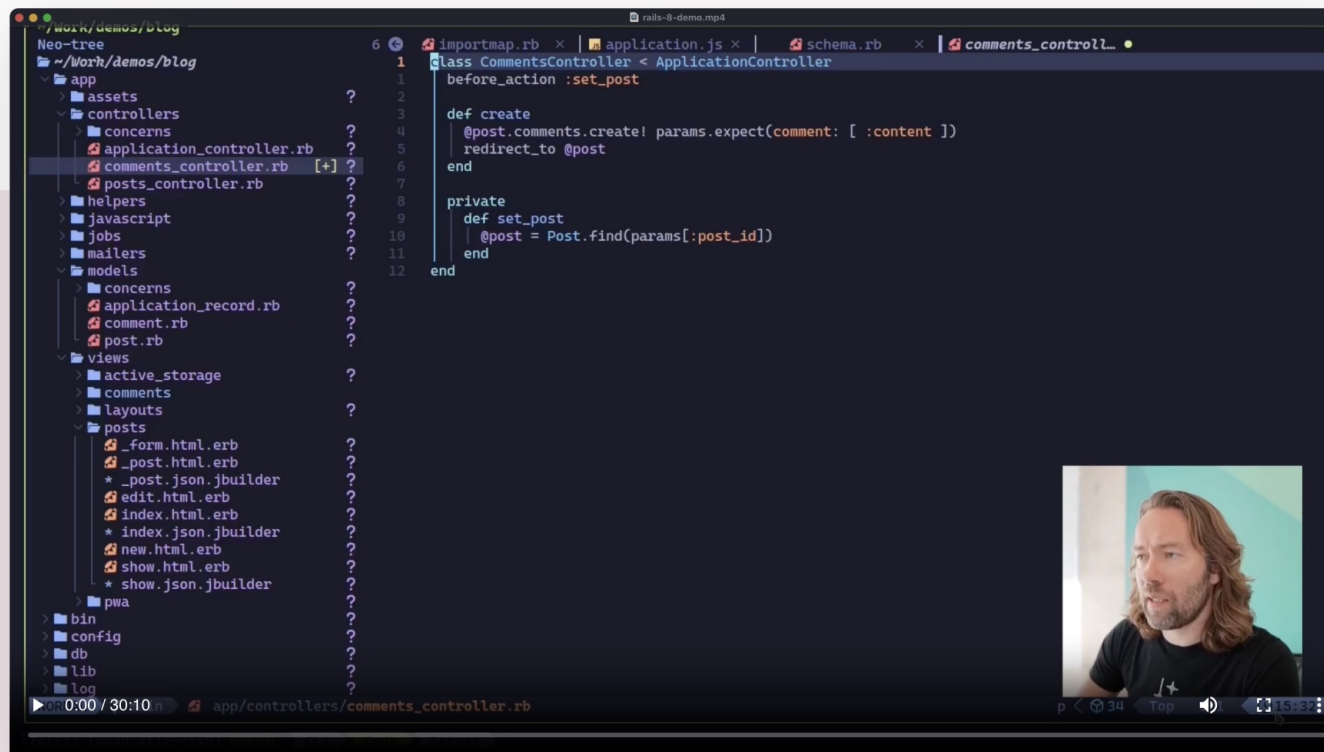
Configure how URLs connect to the controllers using the routing domain language. Routes expose the bundle of actions that go together as a resource: index, show, new, create, edit, update, destroy.

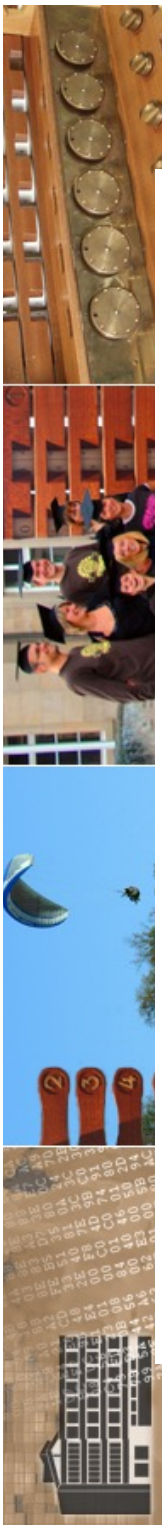


Compress the complexity of modern web apps.

Learn just what you need to get started, then keep leveling up
as you go. **Ruby on Rails scales from HELLO WORLD to IPO.**

Rails 8.1.1 — released October 28, 2025





Ruby on Rails Guides (v8.1.1)

These are the new guides for Rails 8.1 based on [v8.1.1](#). These guides are designed to make you immediately productive with Rails, and to help you understand how all of the pieces fit together.

Start Here

[Getting Started with Rails](#)

Everything you need to know to install Rails and create your first application.

[Install Ruby on Rails](#)

Learn how to install the Ruby programming language and Ruby on Rails.

Models

[Active Record Basics](#)

Active Record allows your models to interact with the application's database. This guide will get you started with Active Record models and persistence to the database.

[Active Record Migrations](#)

Migrations are a feature of Active Record that allows you to evolve your database schema over time. Rather than write schema modifications in pure SQL, migrations allow you to use a Ruby DSL to describe changes to your tables.



Rails Guides are also available for [Kindle](#).



Guides marked with this icon are currently being worked on and will not be available in the Guides Index menu. While still useful, they may contain incomplete information and even errors. You can help by reviewing them and posting your comments and corrections.





Getting Started with Rails

This guide covers getting up and running with Ruby on Rails.

After reading this guide, you will know:

- How to install Rails, create a new Rails application, and connect your application to a database.
- The general layout of a Rails application.
- The basic principles of MVC (Model, View, Controller) and RESTful design.
- How to quickly generate the starting pieces of a Rails application.
- How to deploy your app to production using Kamal.

1 Introduction

Welcome to Ruby on Rails! In this guide, we'll walk through the core concepts of building web applications with Rails. You don't need any experience with Rails to follow along with this guide.

Rails is a web framework built for the Ruby programming language. Rails takes advantage of many features of Ruby so we **strongly** recommend learning the basics of Ruby so that you understand some of the basic terms and vocabulary you will see in this tutorial.

- [Official Ruby Programming Language website](#)
- [List of Free Programming Books](#)

2 Rails Philosophy

Rails is a web application development framework written in the Ruby programming language. It is designed to make programming web applications easier by making assumptions about what every developer needs to get started. It allows you to write less code while accomplishing more than many other languages and frameworks. Experienced Rails developers also report that it makes web application development more fun.



Chapters

1. Introduction

2. Rails Philosophy

3. Creating a New Rails App

- Prerequisites
- Creating Your First Rails App
- Directory Structure
- Model-View-Controller Basics

4. Hello, Rails!

- Autoloading in Development

5. Creating a Database Model

- Database Migrations
- Running Migrations

6. Rails Console

7. Active Record Model Basics

- Creating Records
- Querying Records
- Filtering & Ordering Records
- Finding Records
- Updating Records
- Deleting Records
- Validations

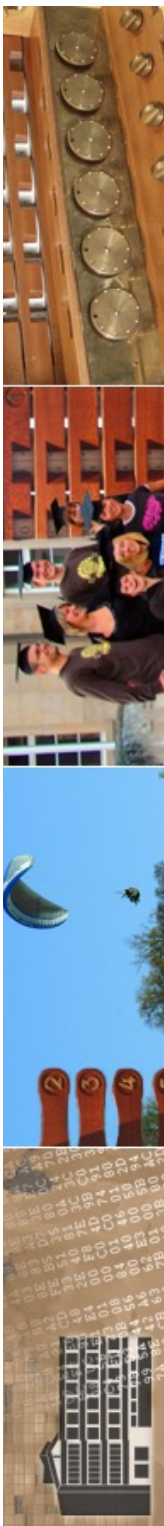
8. A Request's Journey Through Rails

Installation

- zu Installieren sind:
 - aktuelle Ruby-Version
 - es wird mindestens Ruby 3.x benötigt
 - `ruby --version`

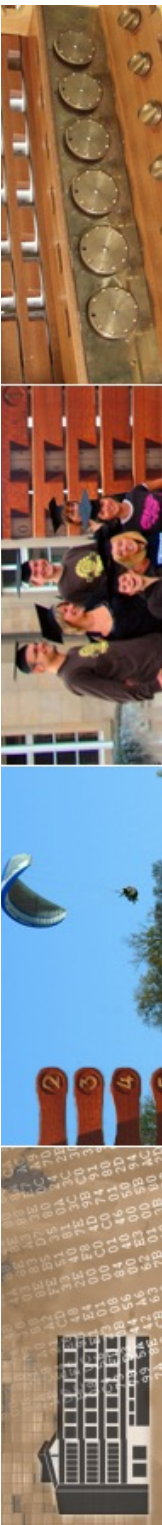
```
thomas@PetitMouton-2 =>
thomas@PetitMouton-2 => ruby --version
ruby 2.6.3p62 (2019-04-16 revision 67580) [universal.x86_64-darwin20]
thomas@PetitMouton-2 =>
thomas@PetitMouton-2 =>
```

- ruby gems
 - dies ist der ruby-rails package manager
- das eigentliche rails-Paket
 - Installation über gems
<https://rubygems.org/>



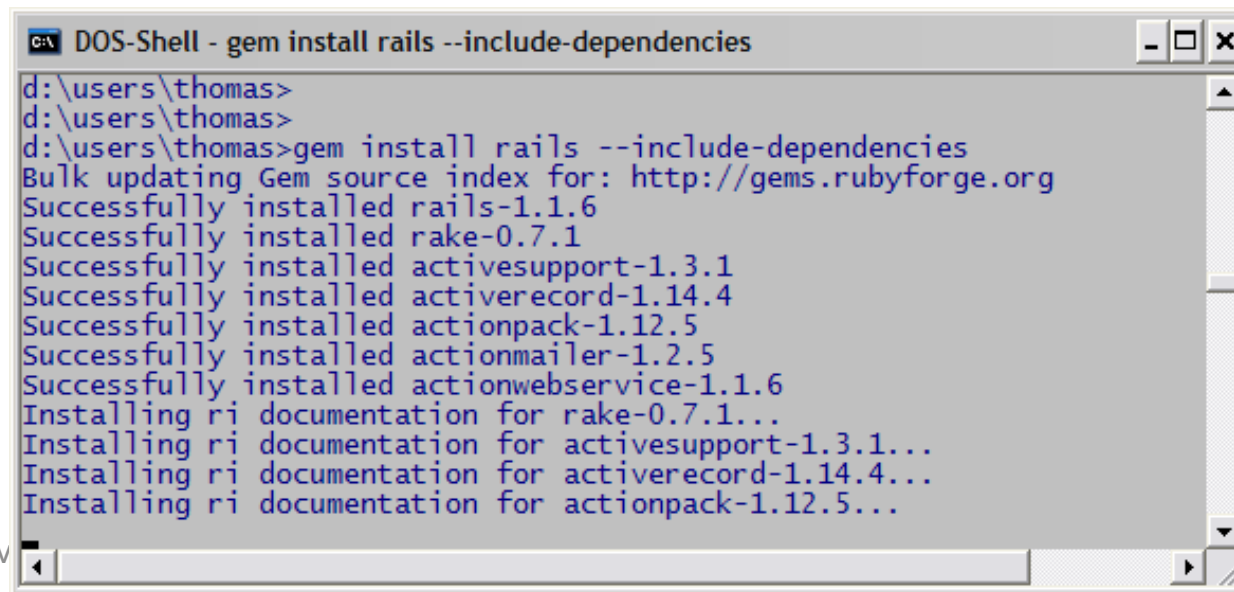
Installation

- was wird noch benötigt:
 - ein Datenbankmanagementsystem
 - Default ist SQLite3



Die eigentliche Rails-Installation

- die eigentliche Installation läuft über gems:
 - `gem install rails --include-dependencies`
- damit alles automatisiert, „frisch“ aus dem Netz



```

DOS-Shell - gem install rails --include-dependencies
d:\users\thomas>
d:\users\thomas>
d:\users\thomas>gem install rails --include-dependencies
Bulk updating Gem source index for: http://gems.rubyforge.org
Successfully installed rails-1.1.6
Successfully installed rake-0.7.1
Successfully installed activesupport-1.3.1
Successfully installed activerecord-1.14.4
Successfully installed actionpack-1.12.5
Successfully installed actionmailer-1.2.5
Successfully installed actionwebservice-1.1.6
Installing ri documentation for rake-0.7.1...
Installing ri documentation for activesupport-1.3.1...
Installing ri documentation for activerecord-1.14.4...
Installing ri documentation for actionpack-1.12.5...
    
```

3.1.3 Installing Rails

To install Rails, use the `gem install` command provided by RubyGems:



```
$ gem install rails
```

Copy

To verify that you have everything installed correctly, you should be able to run the following in a new terminal:



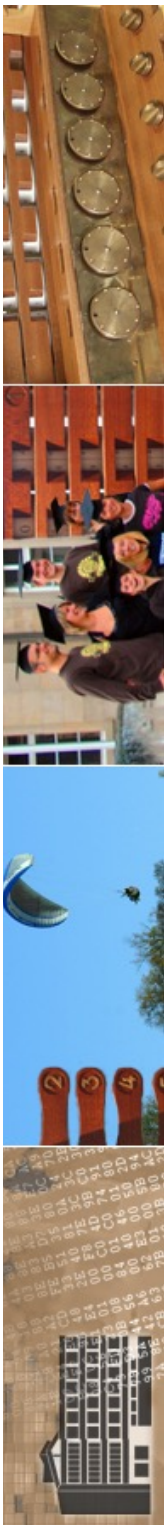
```
$ rails --version
```

Copy

If it says something like "Rails 7.0.0", you are ready to continue.

Erzeugen einer Rails-Applikation

- nach der Installation steht ausführbares Script **rails** zur Verfügung
- damit Anlegen einer Rails-Web-Applikation:
Im Ordner für die Anwendung
 - **rails new <app-name>**
 - etwa: **rails new blog**



3.2 Creating the Blog Application

Rails comes with a number of scripts called generators that are designed to make your development life easier by creating everything that's necessary to start working on a particular task. One of these is the new application generator, which will provide you with the foundation of a fresh Rails application so that you don't have to write it yourself.

To use this generator, open a terminal, navigate to a directory where you have rights to create files, and run:



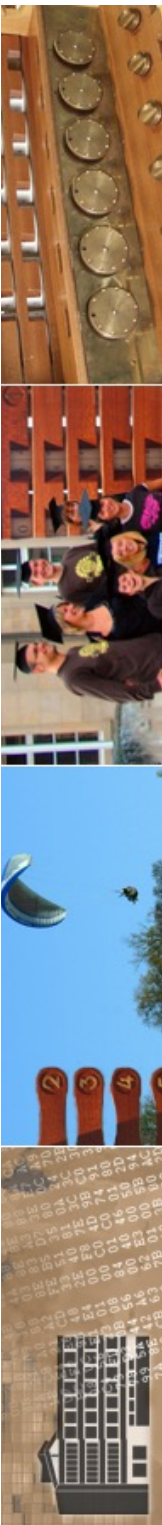
```
$ rails new blog
```

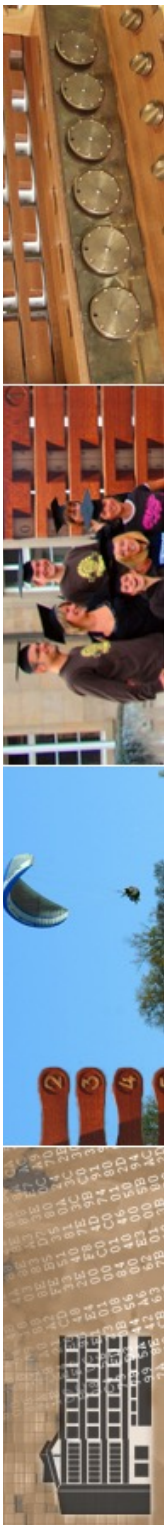
Copy

This will create a Rails application called Blog in a `blog` directory and install the gem dependencies that are already mentioned in `Gemfile` using `bundle install`.

Rails-Verzeichnis der Anwendung

- rails erzeugt das entsprechende Verzeichnis mit zahlreichen Unterordnern, etwa:
 - app: die eigentliche Anwendung, untergliedert in Unterordner models/views/controllers/helpers
 - config: Konfiguration der Datenbank
 - log
 - public: Apache htdocs





File/Folder	Purpose
app/	Contains the controllers, models, views, helpers, mailers, channels, jobs, and assets for your application. You'll focus on this folder for the remainder of this guide.
bin/	Contains the rails script that starts your app and can contain other scripts you use to set up, update, deploy, or run your application.
config/	Contains configuration for your application's routes, database, and more. This is covered in more detail in Configuring Rails Applications .
config.ru	Rack configuration for Rack-based servers used to start the application. For more information about Rack, see the Rack website .
db/	Contains your current database schema, as well as the database migrations.
Gemfile Gemfile.lock	These files allow you to specify what gem dependencies are needed for your Rails application. These files are used by the Bundler gem. For more information about Bundler, see the Bundler website .
lib/	Extended modules for your application.
log/	Application log files.
public/	Contains static files and compiled assets. When your app is running, this directory will be exposed as-is.
Rakefile	This file locates and loads tasks that can be run from the command line. The task definitions are defined throughout the components of Rails. Rather than changing Rakefile, you should add your own tasks by adding files to the lib/tasks directory of your application.
README.md	This is a brief instruction manual for your application. You should edit this file to tell others what your application does, how to set it up, and so on.
storage/	Active Storage files for Disk Service. This is covered in Active Storage Overview .
test/	Unit tests, fixtures, and other test apparatus. These are covered in Testing Rails Applications .
tmp/	Temporary files (like cache and pid files).
vendor/	A place for all third-party code. In a typical Rails application this includes vendored gems.



Starten der Anwendung

- A: mit dem rails-eigenen Webserver WEBrick:
`https://github.com/ruby/webrick`

– im Verzeichnis der Anwendung

`bin/rails server`

- B: mit Apache:

– Konfiguration des Apache



Ruby on Rails: Welcome aboard - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://127.0.0.1:3000/ Google

Umgebungsvariablen in php Ruby on Rails: Welcome aboard



Welcome aboard

You're riding the Rails!

[About your application's environment](#)

Getting started

Here's how to get rolling:

- 1. Create your databases and edit config/database.yml**
Rails needs to know your login and password.
- 2. Use script/generate to create your models and controllers**
To see all available options, run it without parameters.
- 3. Set up a default route and remove or rename this file**
Routes are setup in config/routes.rb.

the Rails site

Join the community

[Ruby on Rails](#)
[Official weblog](#)
[Mailing lists](#)
[IRC channel](#)
[Wiki](#)
[Bug tracker](#)

Browse the documentation

[Rails API](#)
[Ruby standard library](#)
[Ruby core](#)

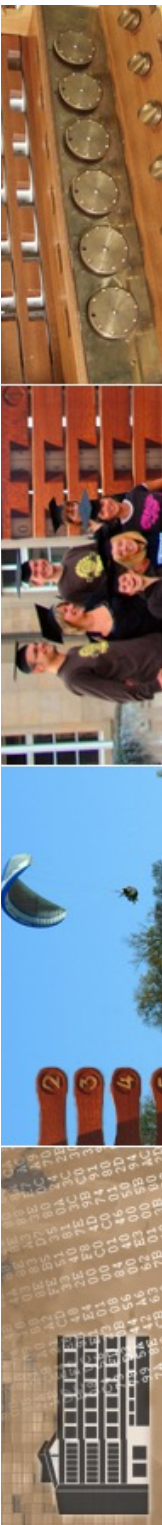
Fertig





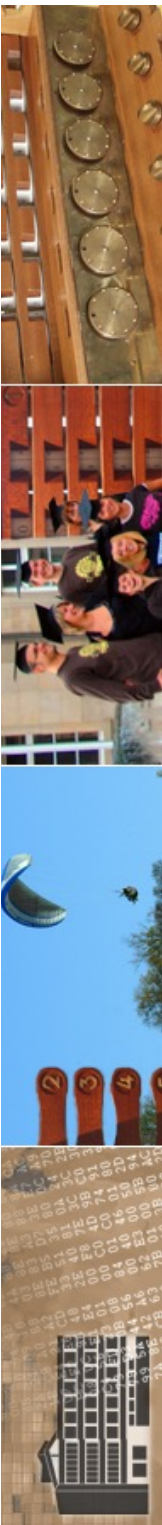
Datenbank

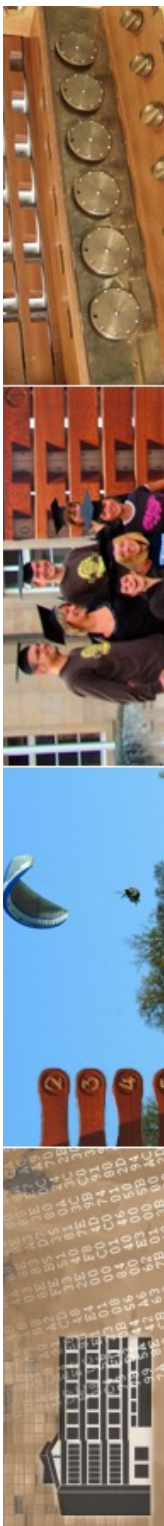
- eine Rails-Anwendung basiert auf einer Datenbank
- Generierung von Anwendungen aus der Datenbank



Anpassen der Konfiguration

- es ist nur die Datenbank-Anbindung zu konfigurieren
- in Unterordner conf, Datei database.yml
- danach Neustart (!) des Webservers





```
1 # MySQL (default setup). Versions 4.1 and 5.0 are recommended.
2 #
3 # Install the MySQL driver:
4 #   gem install mysql
5 # On MacOS X:
6 #   gem install mysql -- --include=/usr/local/lib
7 # On Windows:
8 #   There is no gem for Windows. Install mysql.so from RubyForApache.
9 #   http://rubyforge.org/projects/rubyforapache
10 #
11 # And be sure to use new-style password hashing:
12 #   http://dev.mysql.com/doc/refman/5.0/en/old-client.html
13 development:
14   adapter: mysql
15   database: rails
16   username: thomas
17   password:
18   host: localhost
19
20 # Warning: The database defined as 'test' will be erased and
21 # re-generated from your development database when you run 'rake'.
22 # Do not set this db to the same as development or production.
23 test:
24   adapter: mysql
25   database: test
26   username: thomas
27   password:
28   host: localhost
29
30 production:
31   adapter: mysql
32   database: rails
33   username: thomas
34   password:
35   host: localhost
36
```

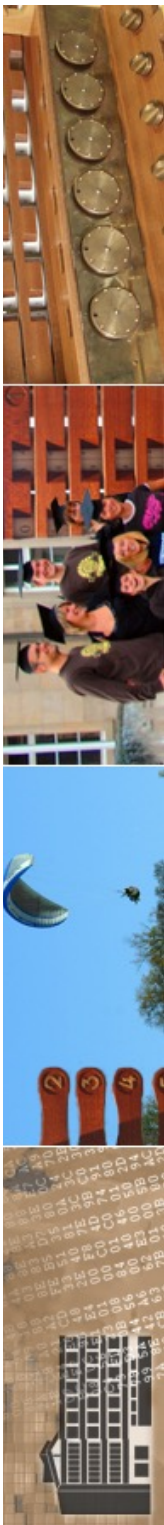
Normal text file nb char : 899 Ln : 1 Col : 1 Sel : 0 UNIX ANSI INS





Erzeugen von M und C

- rails erlaubt ein standardisiertes Erzeugen von Model und Controller über **Generator**
- **Metascript** scripts/generate





C:\ DOS-Shell

```
D:\www\buecher>
D:\www\buecher>ruby script\generate
Usage: script/generate generator [options] [args]

General Options:
  -p, --pretend          Run but do not make any changes.
  -f, --force            Overwrite files that already exist.
  -s, --skip             Skip files that already exist.
  -q, --quiet           Suppress normal output.
  -t, --backtrace       Debugging: show backtrace on errors.
  -h, --help            Show this help message.
  -c, --svn             Modify files with subversion. (Note: svn must be in path)

Installed Generators
  Builtin: controller, integration_test, mailer, migration, model, plugin, scaffold, session_migration, web_service

More are available at http://rubyonrails.org/show/Generators
  1. Download, for example, login_generator.zip
  2. Unzip to directory d:\users\thomas\.rails/generators/login
     to use the generator with all your Rails apps
     or to D:/www/buecher/generators/login
     to use with this app only.
  3. Run generate with no arguments for usage information
     script/generate login

Generator gems are also available:
  1. gem search -r generator
  2. gem install login_generator
  3. script/generate login

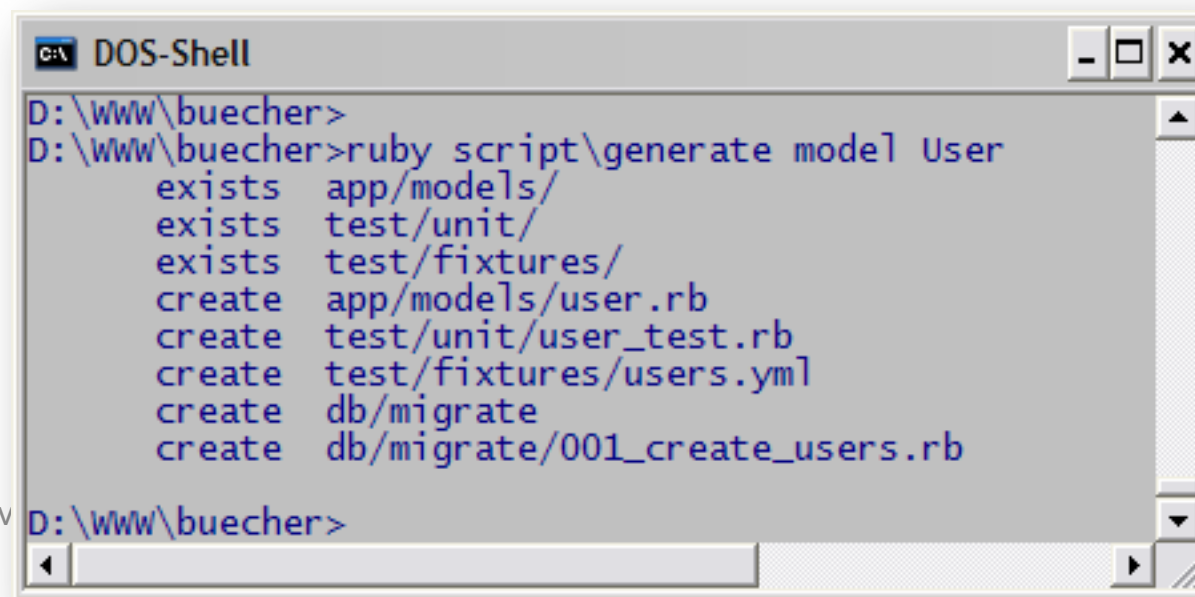
D:\www\buecher>
D:\www\buecher>
```

Controller

- Controller verarbeitet Request
- URL ruft Methode einer Controller-Klasse auf
- Generator:
 - `ruby script/generator controller <Name>`
- Ruby-Klasse, abgeleitet von ApplicationController
- Mapping über `/Name`

Model

- Generierung des Model:
 - zur Datenbanktabelle „users“ gehört das Model „User“
 - „Rails is smart about English pluralization rules, so Company maps to companies, Person maps to people, and so forth.“
 - Erzeugung `ruby script/generate model User`
 - setzt DB-Config und Existenz voraus

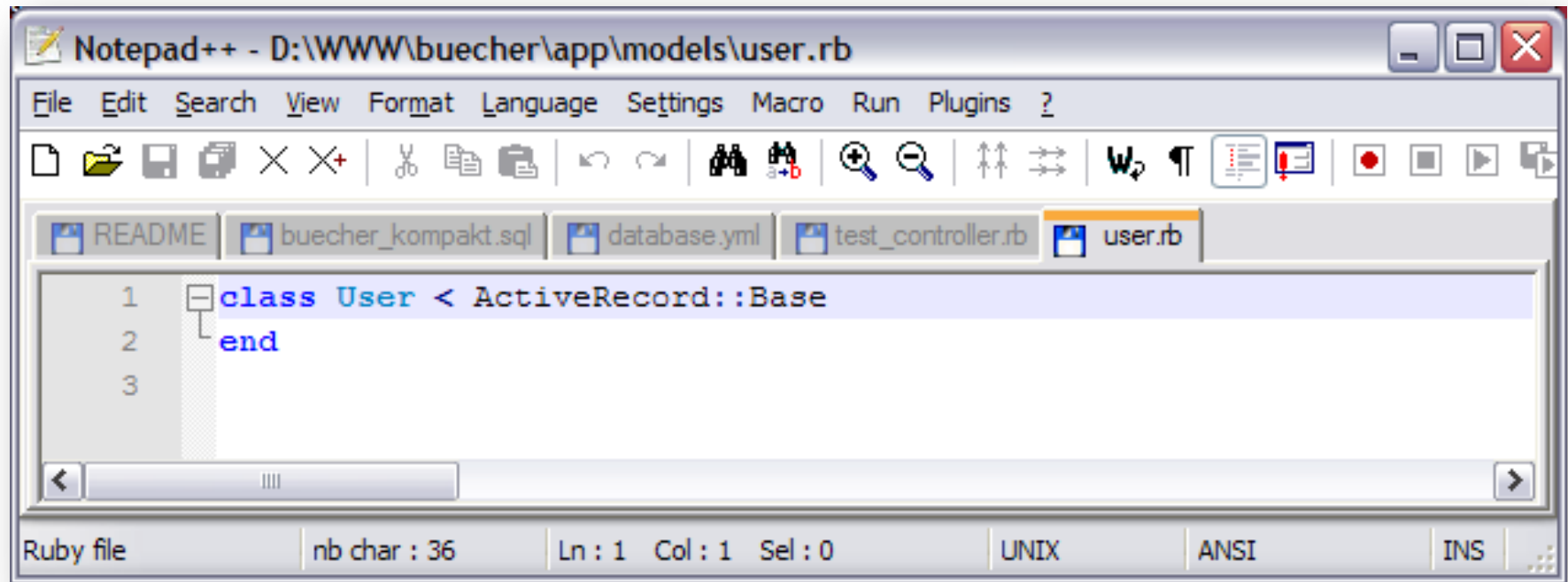


```

C:\ DOS-Shell
D:\www\buecher>
D:\www\buecher>ruby script\generate model User
exists app/models/
exists test/unit/
exists test/fixtures/
create app/models/user.rb
create test/unit/user_test.rb
create test/fixtures/users.yml
create db/migrate
create db/migrate/001_create_users.rb
D:\www\buecher>
    
```

Model

- der Generator erzeugt ein Skeleton einer Ruby-Klasse



The screenshot shows a Notepad++ window titled "Notepad++ - D:\WWW\buecher\app\models\user.rb". The menu bar includes File, Edit, Search, View, Format, Language, Settings, Macro, Run, and Plugins. The toolbar contains various icons for file operations, editing, and navigation. The tab bar shows several open files: README, buecher_kompakt.sql, database.yml, test_controller.rb, and user.rb. The main text area displays the following Ruby code:

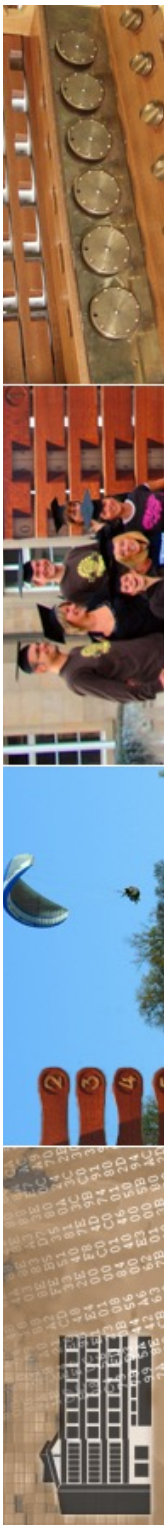
```
1 class User < ActiveRecord::Base
2   end
3
```

The status bar at the bottom indicates "Ruby file", "nb char : 36", "Ln : 1 Col : 1 Sel : 0", and encoding options "UNIX", "ANSI", and "INS".



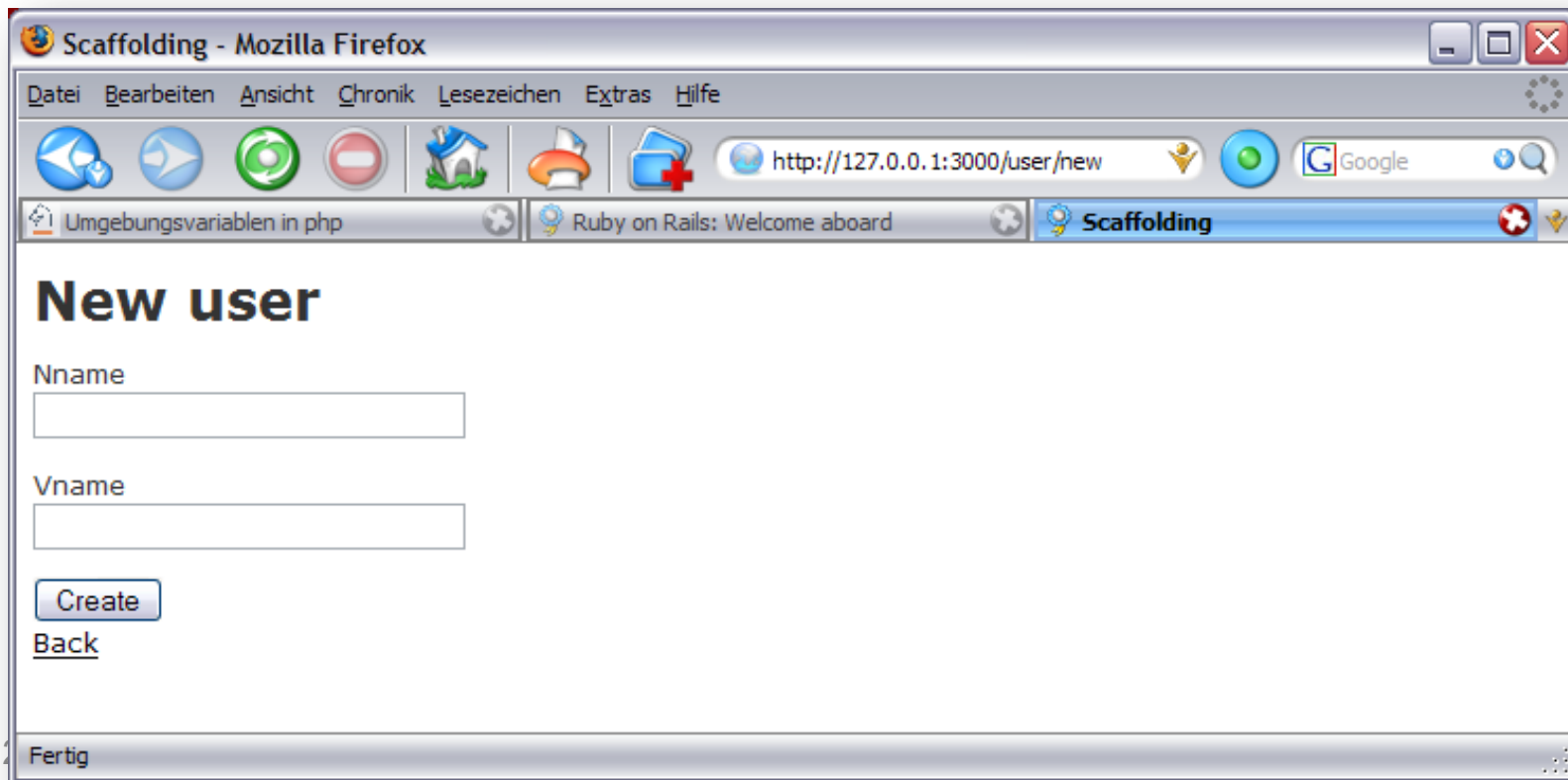
Erzeugung von C und M zusammen

- in den neuesten Rail-Versionen können C und M zusammen erzeugt werden:
 - `ruby script/generate scaffold User`



Standard-Controller für Zugriff

- auf Model soll über Standard-Controller für „CRUD“ zugegriffen werden:
- Create / Read / Update / Delete

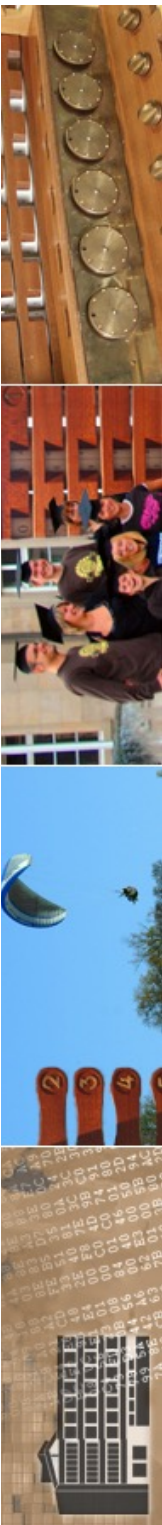


The screenshot shows a Mozilla Firefox browser window titled 'Scaffolding - Mozilla Firefox'. The address bar displays 'http://127.0.0.1:3000/user/new'. The browser's toolbar includes icons for back, forward, home, and search, along with a Google search bar. The page content is titled 'New user' and contains two text input fields labeled 'Nname' and 'Vname'. Below these fields are two buttons: 'Create' and 'Back'. The status bar at the bottom of the browser window shows 'Fertig'.



Mögliche Zugriffe

- `new`
- `list`
- `delete`
- `update`
- `show`
- alles Methoden in Standard-Controller

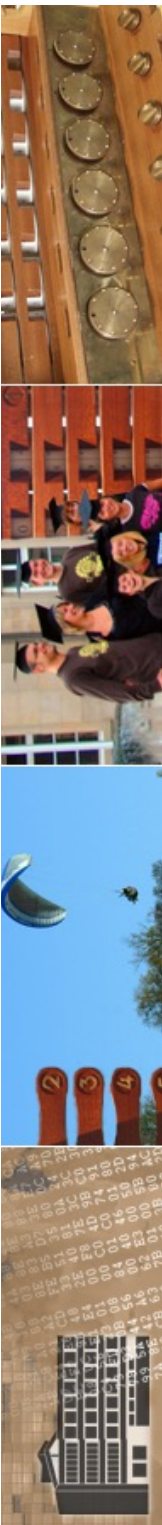




eigene Methoden

- die generierten Methoden (wie list) können überschrieben werden
- dazu muss zunächst nur eine Methode definiert werden

```
def list
  ...code...
end
```



die View: Rails-Templates

- Template enthält im Kern HTML, bereichert um Ruby-Elemente
- ruby-Code in „ASP-Tags“
`<% ruby %>`
 oder
`<%= rubyVariable %>`



app/views/articles/show.html.erb

```
<h1><%= @article.title %></h1>

<%= image_tag @article.cover_image.url %>

<p><%= @article.content %></p>

<%= link_to "Edit", edit_article_path(@article) if Current.user.admin? %>
```

Action Views mix Ruby and HTML.

Templates can use the full versatility of Ruby, excessive code is extracted into helpers, and the domain model is used directly and interwoven with the HTML.



Notepad++ - D:\WWW\buecher\app\views\user\list.rhtml

File Edit Search View Format Language Settings Macro Run Plugins ?

README buecher_kompakt.sql database.yml test_controller.rb user.rb user_controller.rb list.rhtml HelloWorld.php

```

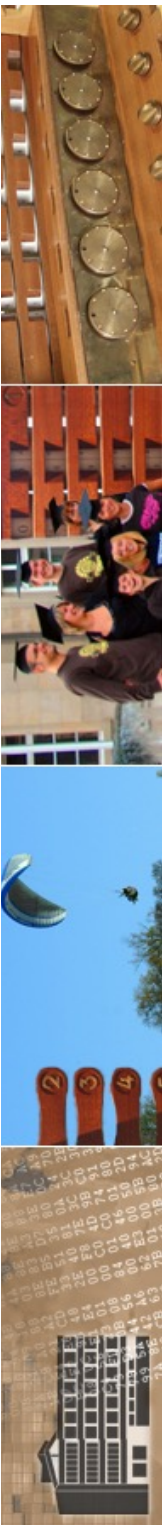
1 <html>
2   <head>
3     <title>WebKompodium: Tabellenausgabe mit Rails</title>
4     <link rel="stylesheet" type="text/css" href="/stylesheets/webkompodium.css">
5   </head>
6   <body>
7
8     <hr><center><h2>Kompodium der Web-Programmierung</h2>
9     <h3>Benutzertabelle mit Rails</h3>
10    <hr>
11    <table>
12      <tr>
13        <th><p align="center">Nachname</b></th>
14        <th><p align="center">Vorname</th>
15      </tr>
16
17      <% @users.each do |user| %>
18        <tr>
19          <td><%= user.nname %></td>
20          <td><%= user.vname %></td>
21        </tr>
22      <% end %>
23    </table>
24    <hr>
25
26  </body>
27 </html>

```

Hyper Text Markup Language file nb char : 557 Ln : 1 Col : 1 Sel : 0 Dos\Windows ANSI INS 0

Mapping

- das zentrale Mapping wird durch die Datei
 - `config/routes.rb`
- festgelegt
- Standard-Situation:
 - `controller-Name/Methoden-Name/ID`



config/routes.rb

```
Rails.application.routes.draw do
  resources :articles do      # /articles, /articles/1
    resources :comments      # /articles/1/comments, /comments/1
  end

  root to: "articles#index" # /
end
```

Action Dispatch route URLs.

Configure how URLs connect to the controllers using the routing domain language. Routes expose the bundle of actions that go together as a resource: index, show, new, create, edit, update, destroy.

Zusammenfassung

das Rails-Framework

- bietet ein „fast-development“ → Prototyping
- direkte Anpassung an das Datenmodell in DB
- MVC automatisch und zwingend
- mit Template-Engine
- automatisches und bequemes Mapping auf Controllermethoden
- minimale Konfiguration



The Rails Doctrine.

David Heinemeier Hansson

English

Ruby on Rails' phenomenal rise to prominence owed much of its lift-off to novel technology and timing. But technological advantages erode over time, and good timing doesn't sustain movements alone over the long term. So a broader explanation of how Rails has continued to not only stay relevant but to grow its impact and community is needed. I propose that the enduring enabler has been and remains its controversial doctrine.

This doctrine has evolved over the past decade, but most of its strongest pillars are also the founding ones. I make no claim to the fundamental originality of these ideas. The chief accomplishment of Rails was to unite and cultivate a strong tribe around a wide set of heretical thoughts about the nature of programming and programmers.

With all that ado, here follows the nine most important pillars of The Rails Doctrine, as perceived by yours truly:

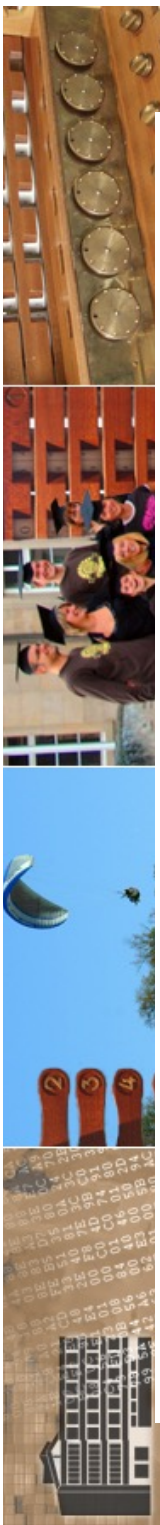
Optimize for programmer happiness

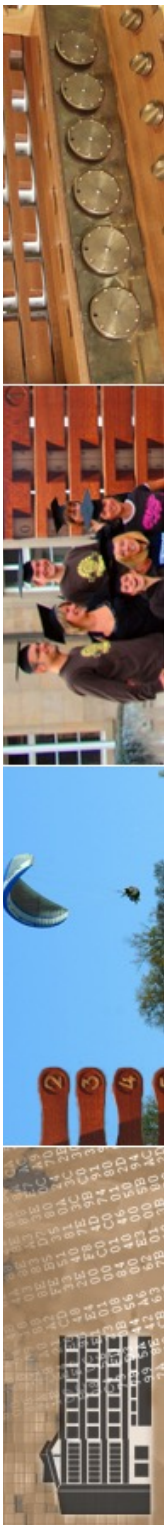
There would be no Rails without Ruby, so it's only fitting that the first doctrinal pillar is lifted straight from the core motivation for creating Ruby.

Ruby's original heresy was indeed to place the happiness of the programmer on a pedestal. Above many other competing and valid concerns that had driven programming languages and ecosystems before it.

Where Python might boast that there's "one and preferably only one way to do something", Ruby relished expressiveness and subtlety. Where Java championed forcefully protecting programmers from themselves, Ruby included a set of sharp knives in the welcome kit. Where Smalltalk drilled a purity of message passing, Ruby accumulated keywords and constructs with an almost gluttonous appetite.

Ruby was different because it valued different things. And most of those things were in service of this yearning for programmer happiness. A pursuit that brought it at odds with not only most other programming environments, but also the mainstream perception of what a programmer was and how they were supposed to act.





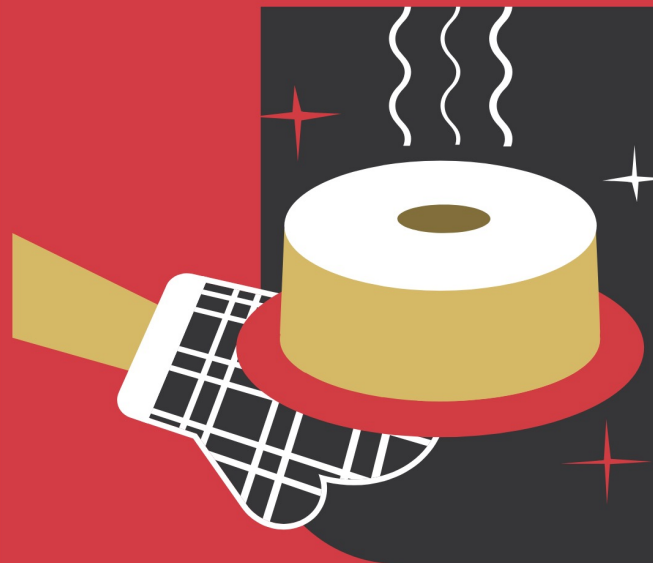


New CakePHP 5.2 Chiffon.
Faster. Simple. Delicious.

 **DOWNLOAD**

 **DOCS**

 **API**



What's new in 5.2

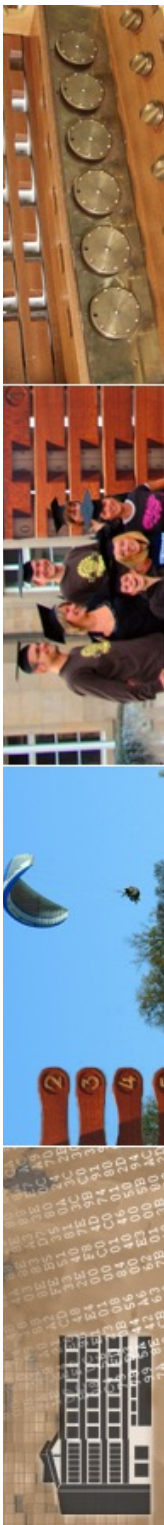
The migration guide has a complete list of what's new in 5.2. We recommend you give that page a read when upgrading. A few highlights from 5.2 are:

- New "counter_cache" command
- New "nativeuuid" database type
- Error handling customization using methods in "ErrorController"
- Helper method for generation importmap for JavaScript files
- Support for PHPUnit 12

LEARN MORE

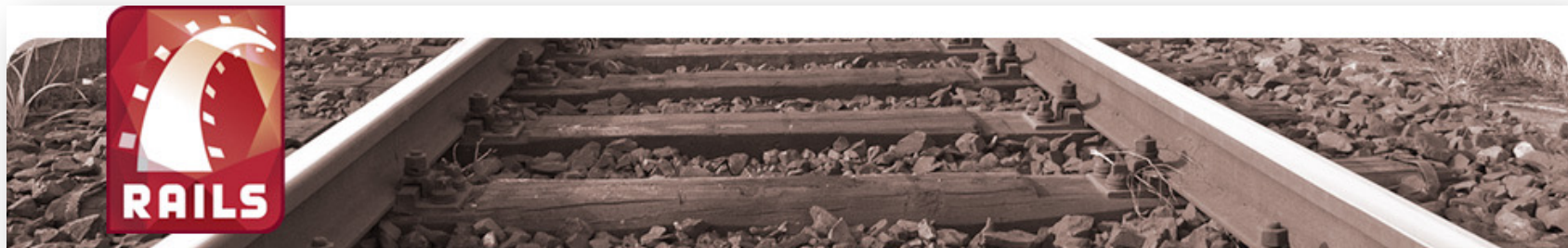
Grundprinzipien von CakePHP

- Grundprinzipien von CakePHP sind weitgehend von Ruby on Rails übernommen
 - **DRY:** don't repeat yourself
Vermeidung von Redundanzen
 - **Rapid Development**
kurze Planungsphase, schnell ausführbarer Code
 - **CRUD:** create, read, update, and delete
direkte Erstellung dieser vier DB-Operationen
 - **Convention over Configuration**
Konventionen ersetzen Konfigurationen



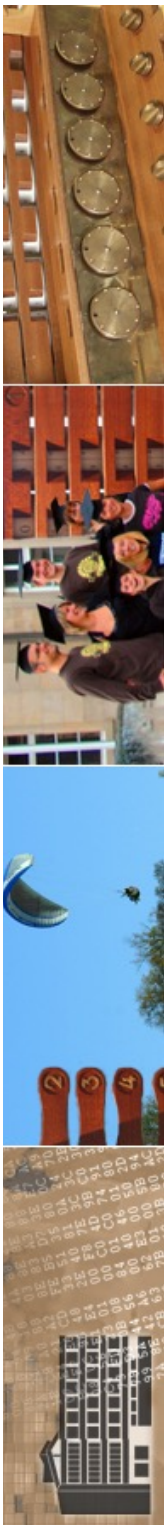
die Idee von CakePHP

- CakePHP ist stark an Ruby on Rails angelehnt
 - "Rapid Development"



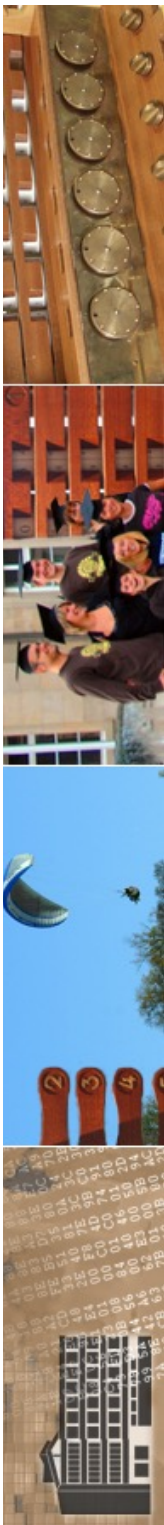
CakePHP Komponenten

- CakePHP verfügt über typische Eigenschaften und Komponenten:
 - MVC in CakePHP
 - Plugins
 - Routes und Dispatcher
 - CakePHP Konsole



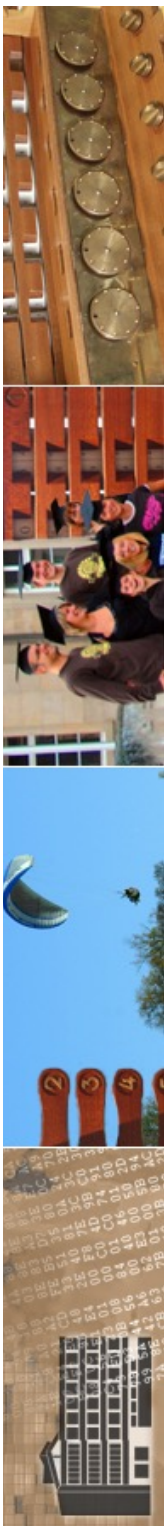
Entwicklung einer CakePHP-Anwendung

- Schritte zur CakePHP-Applikation
 - Installation CakePHP
 - Datenbankschema
 - Model generieren
 - Controller erstellen
 - Views erstellen
 - Anwendung optimieren
- wichtig: Rolle der Datenbank/des Models



Installation

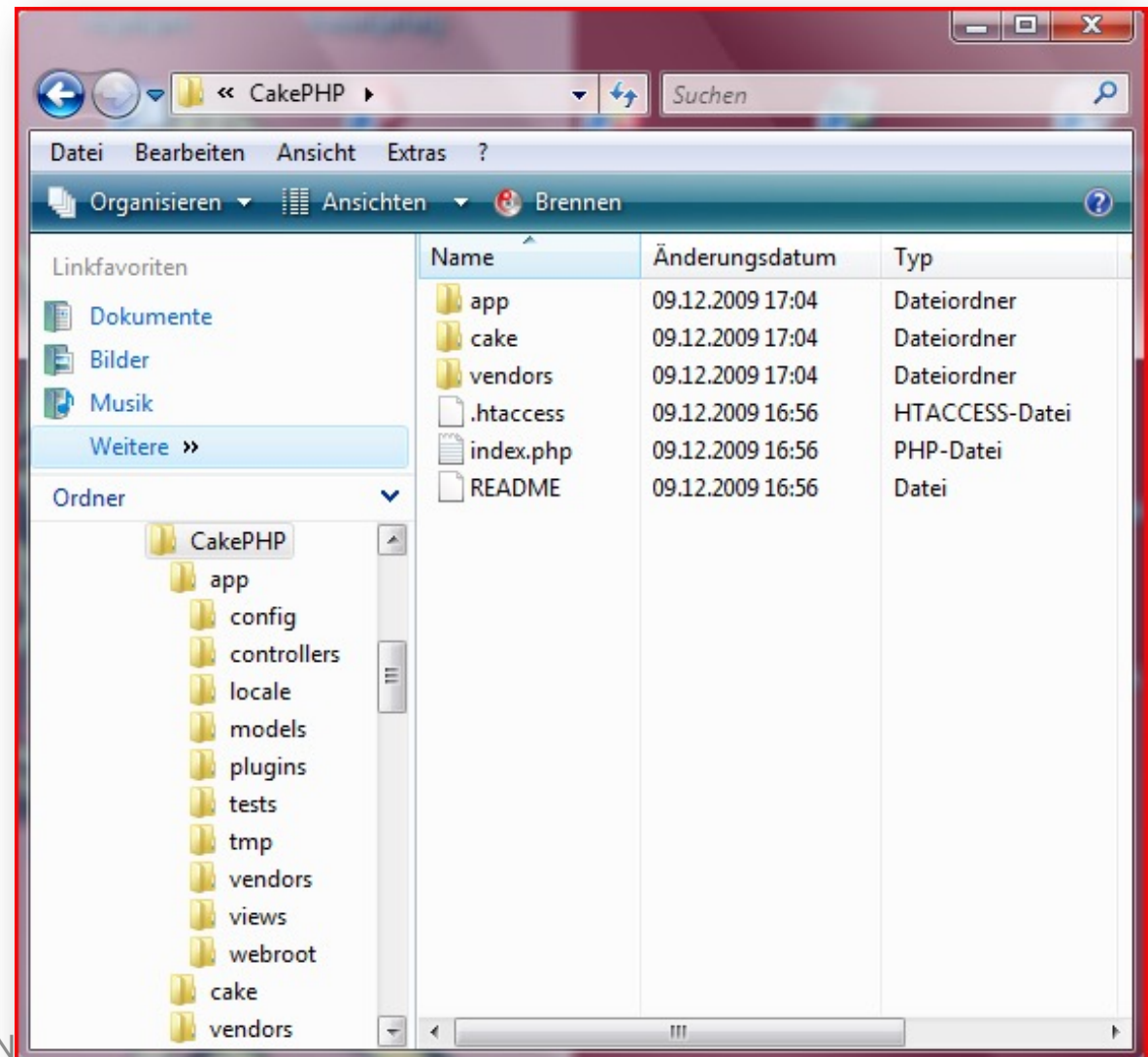
- außer den Framework-Sourcen werden benötigt:
 - Web-Server (Apache 2.x)
 - aktuelles PHP-Modul
 - DBMS (MySQL/MariaDB, auch Postgres, SQLite, Oracle, MS SQL, ...)
- CakePHP: Sourcen-Archiv von <http://cakephp.org>
 - Entpacken in htdocs
- wichtig: Apache-Direktive **AllowOverride all**
 - weil Konfiguration in .htaccess-Dateien



Verzeichnisstruktur

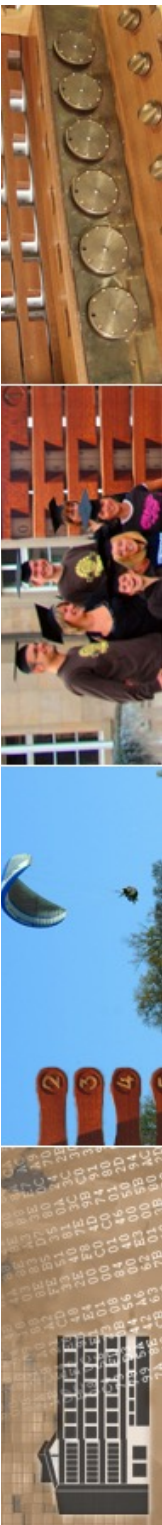
nach Entpacken
typische
Verzeichnis-
struktur

- Ordner
app
mit wichtigen
Unterordnern



Unterordner des Ordners app

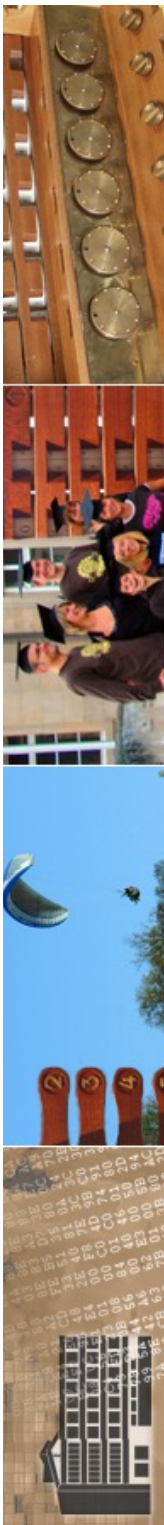
- config
 - Konfigurationsdateien, etwa DB-Paramater
- controllers
 - Verzeichnis der Controller
- locale: Internationalisierung
- models
- plugins
 - CakePHP-Plugin-Packages
- webroot
 - Documentroot
- views





die Datenbankkonfiguration

- im Unterordner `app/conf` liegt die Datei `database.php`
- diese enthält die Datenbankparameter





```

D:\www\CakePHP\app\config\database.php - Notepad++
Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen
Fenster ?
index.php IndexController.php BeispielController.php index - Kopie.php index.phtml ErrorController.php Db
79 class DATABASE_CONFIG {
80
81     var $default = array(
82         'driver' => 'mysql',
83         'persistent' => false,
84         'host' => 'localhost',
85         'login' => 'thomas',
86         'password' => '',
87         'database' => 'webkompendium',
88         'prefix' => '',
89     );
90
nb char : 3312 nb line : 101 Ln : 78 Col : 2 Sel : 0 UNIX ANSI INS

```





CakePHP: the rapid devel... x

← → ↻ ☆ http://127.0.0.1:81/cakePHP/ ▶ 📄 ⚙

CakePHP: the rapid development php framework

Release Notes for CakePHP 1.2.4.8284.

[Read the changelog](#)

[Notice \(1024\)](#): Please change the value of 'Security.salt' in app/config/core.php to a salt value

Your tmp directory is writable.

The *FileEngine* is being used for caching. To change the config edit APP/config/core.php

Your database configuration file is present.

Cake is able to connect to the database.

Editing this Page

To change the content of this page, create: APP/views/pages/home.ctp.
To change its layout, create: APP/views/layouts/default.ctp.
You can also add some CSS styles for your pages at: APP/webroot/css.

Getting Started

[new CakePHP 1.2 Docs](#)

[The 15 min Blog Tutorial](#)



die Cake-Konsole

```

DOS Shell
Welcome to CakePHP v1.2.4.8284 Console
-----
Current Paths:
-app: console
-working: D:\www\CakePHP\cake\console
-root: D:\www\CakePHP\cake
-core: D:\www\CakePHP

Changing Paths:
your working path should be the same as your application path
to change your path use the '-app' param.
Example: -app relative/path/to/myapp or -app /absolute/path/to/myapp

Available Shells:

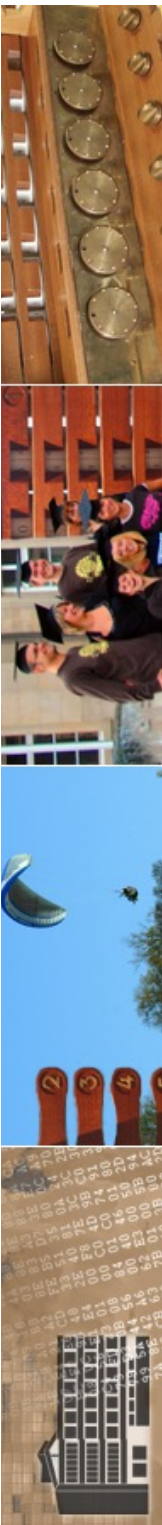
D:\www\CakePHP\vendors\shells:
- none

CORE\console\libs:
    acl
    api
    bake
    console
    i18n
    schema
    testsuite

To run a command, type 'cake shell_name [args]'
To get help on a specific command, type 'cake shell_name help'
  
```

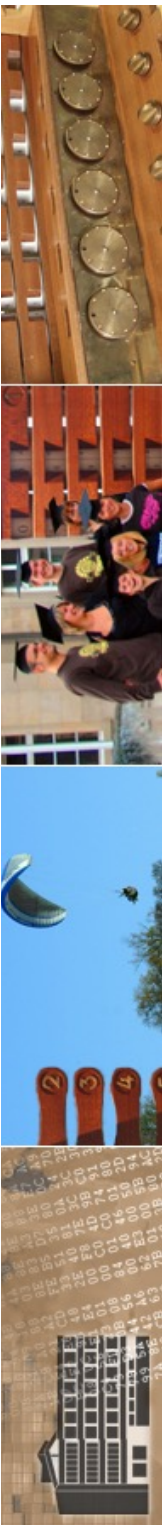
Einsatz der Konsole

- die CakePHP-Konsole dient
 - zum Testen
 - zum Generieren von Code



erstellen einer Applikation

- Ausgangspunkt ist eine Datenbanktabelle
 - diese definiert Model
 - `mysql> create table students(mtknr integer not null, nname varchar(64) not null, vname varchar(64), fach varchar(64));`
 - Query OK, 0 rows affected (0.06 sec)
 - `mysql> insert into students (mtknr, nname, vname, fach) values (1,'Chagall', 'Marc', 'Medieninformatik');`
 - Query OK, 1 row affected (0.01 sec)



die Model-Klasse

- die Model-Klasse nimmt Bezug zur bestehenden Datenbanktabelle

```

1 <?php
2     // Grundlagen der Web-Entwicklung
3     // CakePHP: einfache Model-Klasse
4
5     class Student extends AppModel {
6         var $name='Student';
7     } // class
8     ?>
  
```



die Controller-Klasse

minimale Controller-Klasse

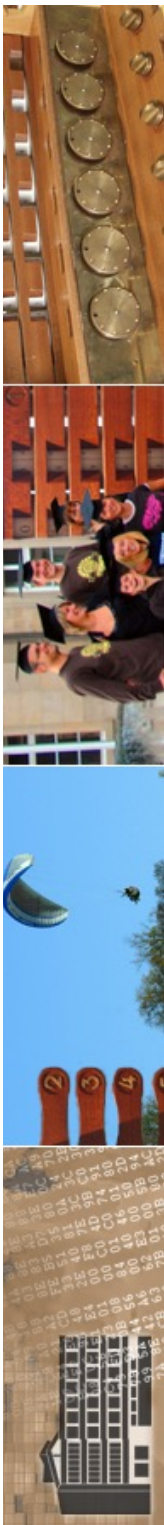
```

1 <?php
2 // Grundlagen der Web-Entwicklung
3 // CakePHP: einfache Controller-Klasse
4
5 class StudentsController extends AppController {
6     var $name="Students";
7     var $scaffold;
8 } // class
9 ?>
  
```

The screenshot shows a Notepad++ window with the file path 'D:\www\CakePHP\app\controllers\students_controller.php'. The code defines a 'StudentsController' class that extends 'AppController'. It includes two public properties: '\$name' set to 'Students' and '\$scaffold'. The code is syntax-highlighted and includes line numbers from 1 to 9.

Funktionalität dieser Anwendung

- bereits mit diesem geringen Aufwand hat die Anwendung eine umfassende Funktionalität
- Hintergrund: Scaffolding: CRUD-Funktionalität
 - durch Flag-Variable `$scaffold` im Controller aktiviert
 - Methoden: index, view, add, edit, delete





CakePHP: the rapid development php framework

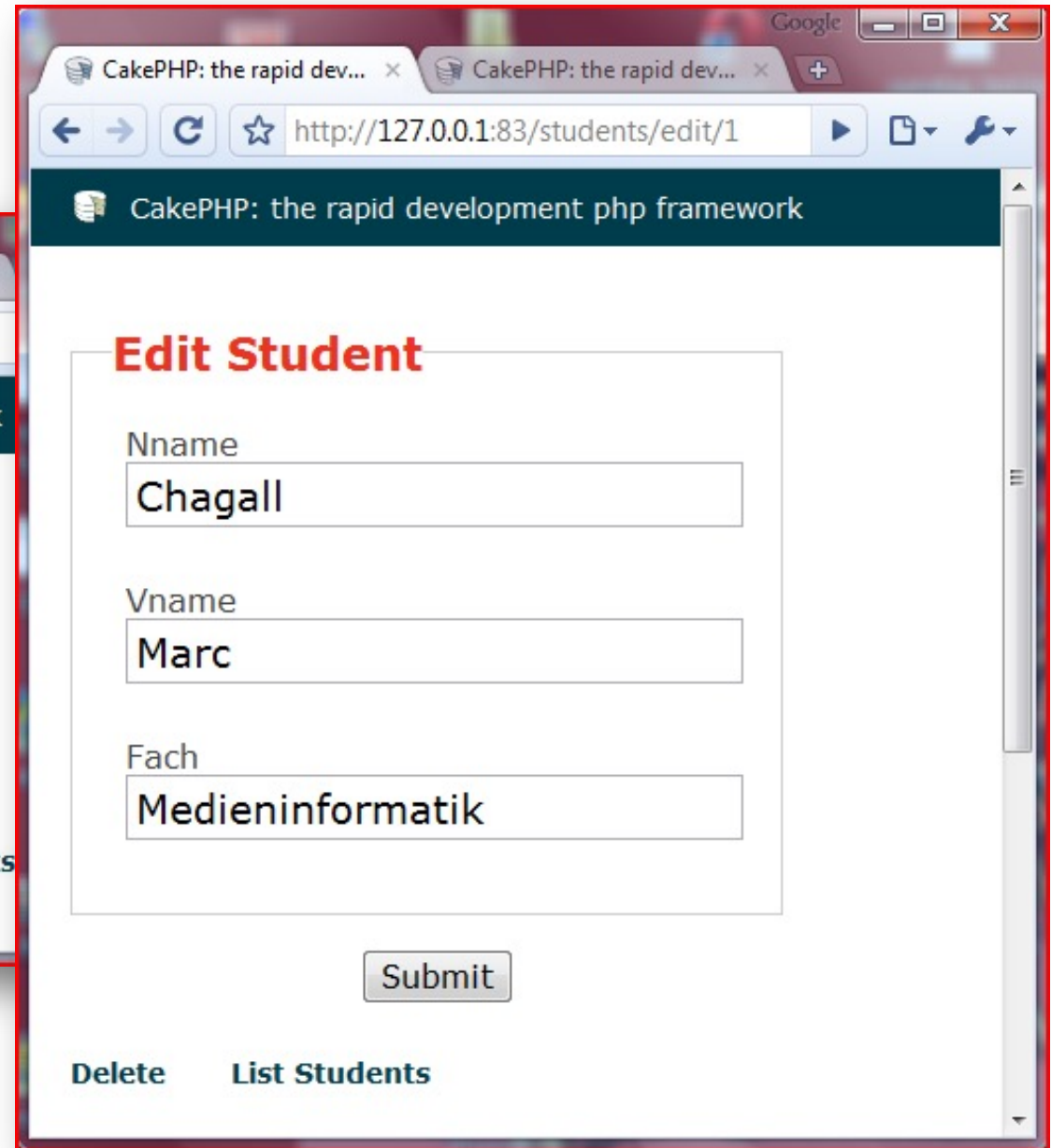
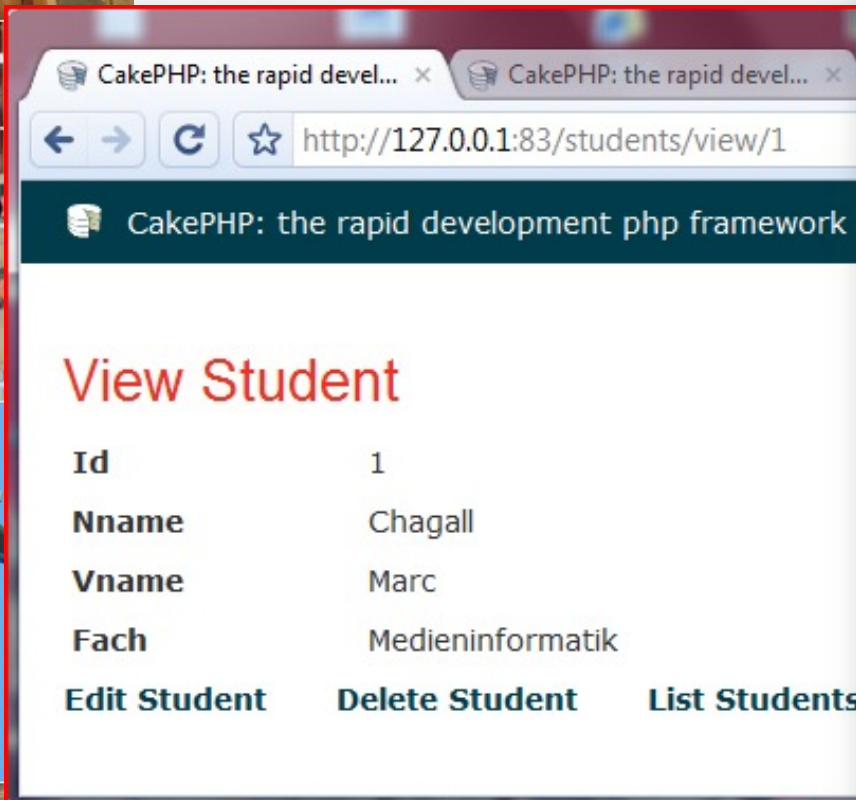
Students

Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1

Id	Nname	Vname	Fach	Actions
1	Chagall	Marc	Medieninformatik	View Edit Delete

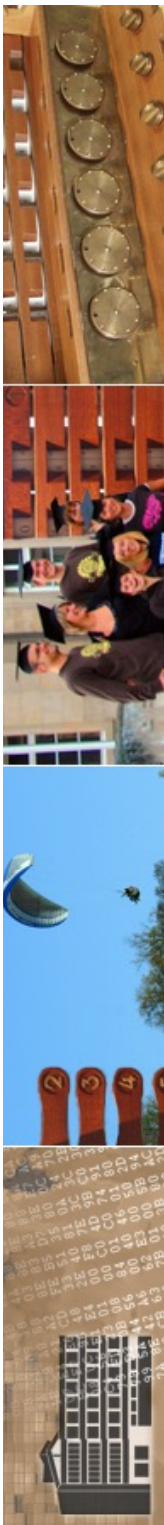
<< previous | next >>

New Student



Konfiguration

- im Unterordner `app/config` befinden sich die Konfigurationsdateien des Frameworks
- zentrale Bedeutung
 - `core.php` : Hauptkonfiguration
 - `database.php` : Datenbankzugriff
 - `routes.php` : Routing

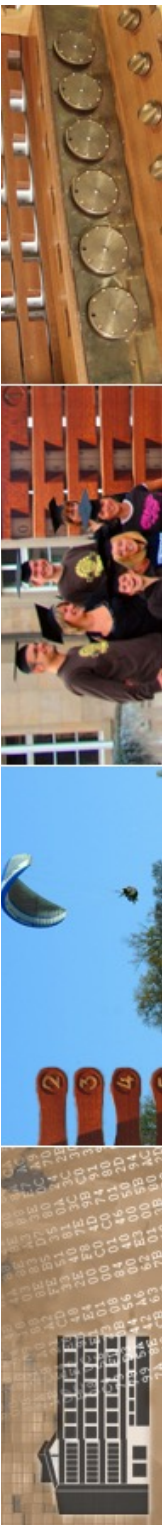


der Controller

- natürlich auch möglich: direktes Implementieren der Methoden *ohne* Scaffolding
- Werteübergabe: Prinzipaufbau der URL

`http://domaine/controller/action/parameter1/parameter2`

– **REST**: Representational State Transfer





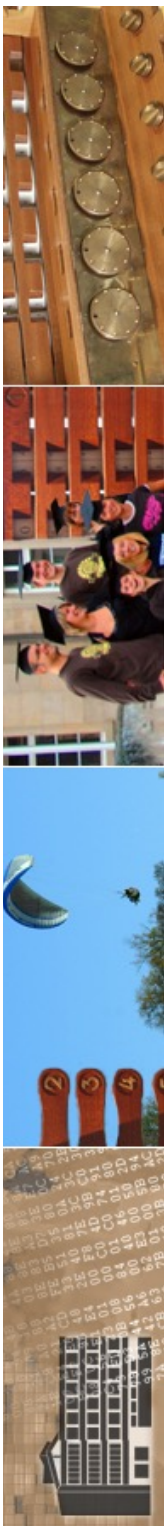
einfache Methode

- Ergänzung einer einfachen Methode:

```
function index() {  
  
    $students = $this->Student->find('all',  
                                     array('order' => 'created DESC'));  
  
    $this->set('students', $students);  
}
```

Konventionen

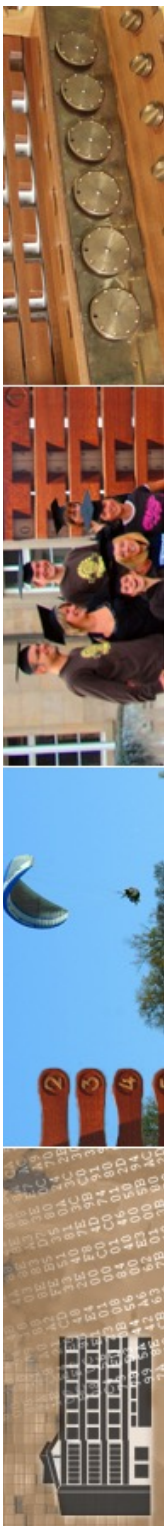
- CakePHP verwendet vielfältig Konventionen
 - Dateinamen: Kleinbuchstaben, Worttrennung durch _
 - Model und DB-Tabelle: gleiche Bezeichnung, Model im Singular, DB-Tabelle im Plural
 - students ⇔ Student
 - our_students ⇔ OurStudent



View

- Template-Engine: HTML-Dateien mit PHP-Code
 - Dateien .ctp : Cake Templates
 - Konvention:
Template zu Methode view in Controller
StudentsController:

`.../views/students/view.ctp`





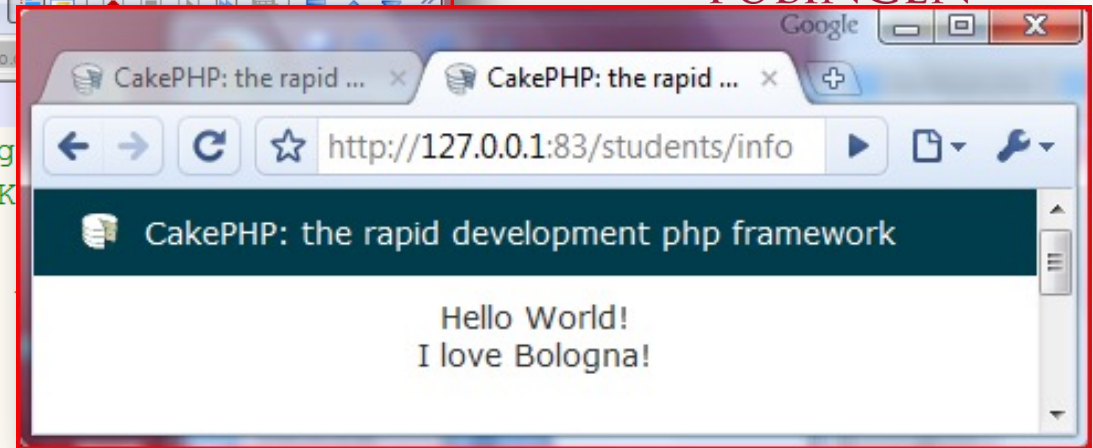
```
D:\www\CakePHP\app\controllers\students_controller.php - Notepad++

Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ? X

database.php student.php htaccess index.php students_controller.php info.

1 <?php
2 // Grundlagen der Web-Entwicklung
3 // CakePHP: einfache Controller-K
4
5 class StudentsController extends
6
7     var $name="Students";
8     var $scaffold;
9     var $message;
10
11 function info() {
12     //$message = "Hallo Marmels!";
13     $this->set('message','I love Bologna!');
14 } // function
15
16 } // class
17 ?>
```

PHP Hyp nb char: 347 nb line: 17 Ln: 1 Col: 1 Sel: 0



```
D:\www\CakePHP\app\views\students\info.ctp - Notepad++

Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX
Erweiterungen Fenster ? X

database.php student.php htaccess index.php students_controller.php info.ctp

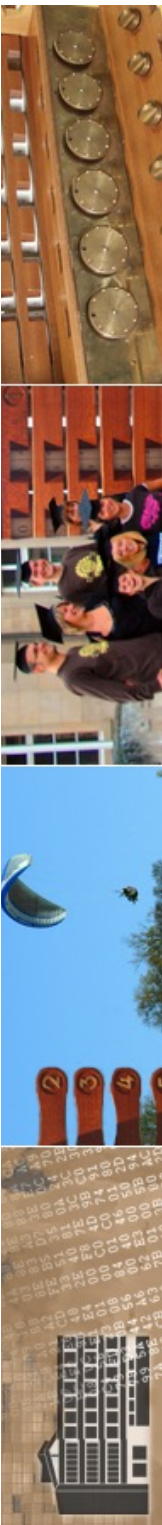
1 <center>
2     <?php
3         echo "Hello World!<BR />";
4         echo $message;
5     ?>
6 </center>
```

nb char: 79 nb line: 6 Ln: 1 Col: 1 Sel: 0 UNIX ANSI INS



Meta-Templates

- das Template enthält "nur den body"
- Layouts: Rahmen
- abgelegt in `.../cake/libs/view/layouts`
- Standard ist `default.ctp`





D:\www\CakePHP\cake\libs\view\layouts\default.ctp - Notepad++

Datei Bearbeiten Suchen Ansicht Format Sprachen Einstellungen Makro Ausführen TextFX Erweiterungen Fenster ?

database.php student.php students_controller.php htaccess index.php info.ctp core.php httpd-vhosts.conf default.ctp

```

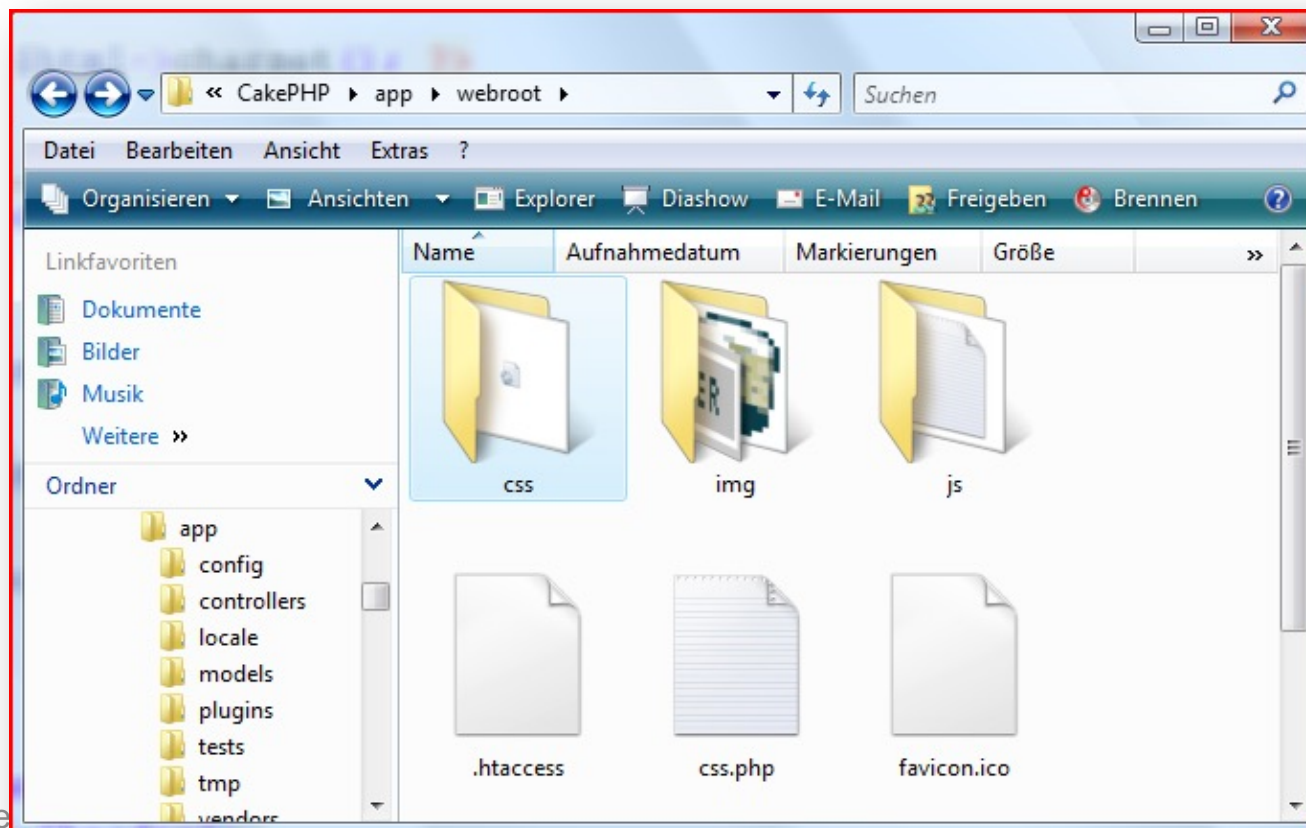
25 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w
26 <html xmlns="http://www.w3.org/1999/xhtml">
27 <head>
28     <?php echo $html->charset(); ?>
29     <title>
30         <?php __('CakePHP: the rapid development php framework:'); ?>
31         <?php echo $title_for_layout; ?>
32     </title>
33     <?php
34         echo $html->meta('icon');
35
36         echo $html->css('cake.generic');
37
38         echo $scripts_for_layout;
39     ?>
40 </head>
41 <body>
42     <div id="container">
43         <div id="header">
44             <h1><?php echo $html->link(__('CakePHP: the rapid development ph
45         </div>
46         <div id="content">

```

Hyper Text Markup Language file nb char: 1872 nb line: 64 Ln:1 Col:1 Sel:0 UNIX ANSI INS

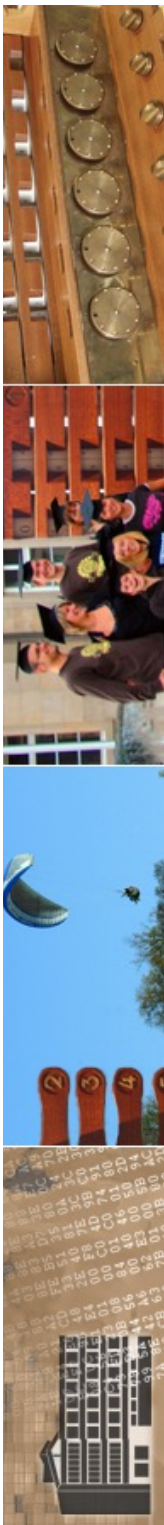
Ordner webroot

- Standard-Ziel aller Anfragen ist defaultmäßig der Ordner `app/webroot`
 - dort entsprechende Unterordner



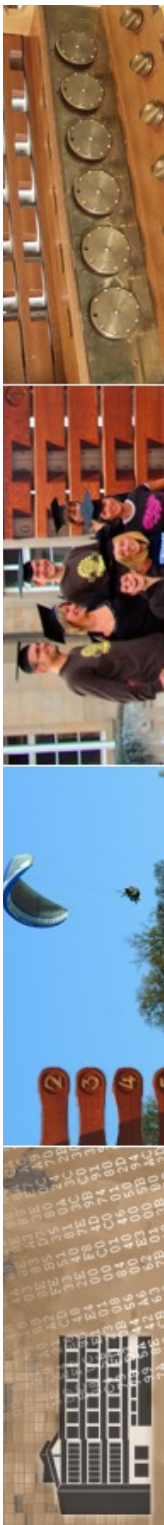
zahlreiche Möglichkeiten

- CakePHP bietet zahlreiche Möglichkeiten, die View zu erstellen
 - Helper-Klassen: Kapseln Code in Views und Layouts, der in mehreren Ansichten verwendet wird: Vermeidung von doppeltem Code
 - Steuerung über `$html`, etwa `$html->css(...)`
 - **JavaScript und Ajax**
 - ...u.v.m. ...





Laravel





Laravel

Products ▾

Framework ▾

Resources ▾

Docs

Partners ↗

82K

Search docs 🔍



Build and ship software with tools crafted for productivity

Laravel provides a complete ecosystem for web artisans. Our open source PHP framework, products, packages, and starter kits offer everything you need to build, deploy, and monitor web applications.

Get started

Start deploying

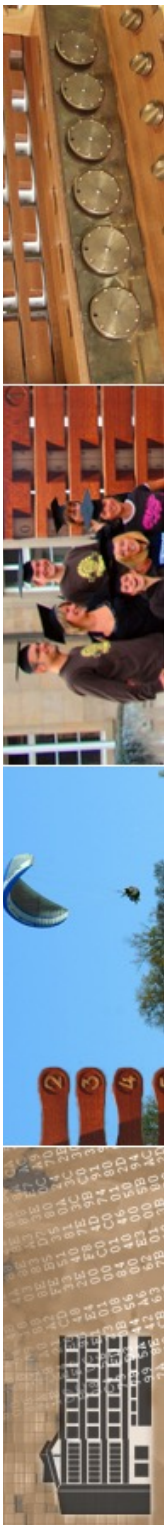
```
> composer global require laravel/installer
```





Das PHP-Framework Laravel

- modernes und beliebtes PHP-Framework, das die Entwicklung von Webanwendungen deutlich vereinfacht. Es setzt auf eine klare, elegante Syntax und bringt viele nützliche Funktionen direkt mit - etwa Routing, Datenbank-Anbindung, Authentifizierung, Warteschlangen, Caching und ein durchdachtes MVC-Architekturmodell
- eine strukturierte, effiziente und entwicklerfreundliche Basis, um robuste Webanwendungen schneller und sauberer zu bauen



Eloquent-ORM

- Eloquent ist Laravels Object-Relational-Mapper (ORM), der es ermöglicht, Datenbanken mit PHP-Objekten zu interagieren

Eloquent ORM

Introduction

The Eloquent ORM included with Laravel provides a beautiful, simple ActiveRecord implementation for working with your database. Each database table has a corresponding "Model" which is used to interact with that table.

Before getting started, be sure to configure a database connection in `config/database.php`.

...und nun...

- wir haben einen Klassiker unter den modernen Frameworks kennen gelernt:
Ruby On Rails
- und mit CakePHP
die Variante „in PHP“
- als nächstes: Web-Services und Rechtliches

