



# MODUL 133

## TEIL 1: RAILS SERVER FÜR DIE ENTWICKLUNG IN BETRIEB NEHMEN

Ralph Maurer

# Inhaltsverzeichnis AB133-01

## Modul 133: Webapplikationen mit Session-Handling realisieren

### Teil 1: Rails Server für die Entwicklung in Betrieb nehmen

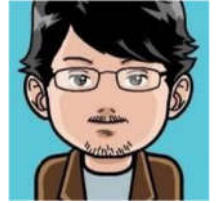
Inhaltsverzeichnis AB133-01 .....	1
Was ist Ruby on Rails? .....	2
Ruby .....	2
Rails.....	2
Installation von Ruby on Rails auf Ubuntu .....	2
Erstellen des Workspace Ordners .....	6
Erstellen einer neuen Rails App .....	7
Praktische Arbeit: Beim Systemstart den Rails-Server starten .....	9



# Was ist Ruby on Rails?

## Ruby

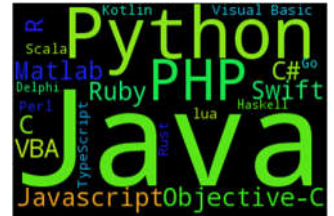
Ruby ist eine Programmiersprache. Sie wurde vor ca. 20 Jahren von Yukihiro "Matz" Matsumoto erfunden.



Matz

Ruby ist nach den Spitzenreitern Java und Python eine der beliebtesten Programmiersprachen der Welt. Die Popularität von Ruby basiert auf dem Erfolg in der Webprogrammierung (besonders durch das Framework Rails).

Matz betont jedoch immer wieder, dass Ruby eine allgemeine Sprache ist wie Java und C# und nicht auf Webentwicklung reduziert werden darf.



Quelle:

<http://pypl.github.io/PYPL.html>

## Rails

Rails ist eine Softwarebibliothek, die Ruby speziell für die Webprogrammierung erweitert. Rails wurde von dem Unternehmer und Autorennfahrer David Heinemeier Hansson entwickelt. David nannte seine Entwicklung "Ruby on Rails", heute spricht man meist nur noch von "Rails". Aktuell ist er Inhaber der Softwarefirma Basecamp.



David Heinemeier Hansson

Technisch handelt es sich bei Rails um eine Paketbibliothek (genauer eine RubyGem), die über die Befehlszeilenschnittstelle des Betriebssystems installiert wird. RubyGem ist ein Paketmanager, der es erlaubt, Ruby-Programme als Standardprogramme zu verteilen. Diese standardisierten, selbstenthaltenden und eigenständigen Rubyprogramme nennt man Gem (in Deutsch übersetzt «Edelstein»).

## Installation von Ruby on Rails auf Ubuntu

Vorbemerkung: RailsGirls (<http://railsgirls.com/>) ist eine Web-Community für Frauen mit Anleitungen und Tools für Rails und Ruby on Rails. RailsGirls ist besonders für Einsteiger hilfreich und wird durchaus auch von Männern benutzt ;-)

Viele Rails Entwicklerinnen und Entwickler wie zum Beispiel die RailsGirls (<http://railsgirls.com/>) nutzen unixoide OS, da diese besonders gut geeignet sind.



Unter <https://gorails.com/setup> gibt es Anleitungen für die Installation von Ruby on Rails auf unterschiedlichen Plattformen. Unixoide OS (d.h. Unix-ähnliche Systeme) sind besonders gut geeignet für Rails.

- ● ● Anstatt Ruby on Rails auf einem Computer zu installieren, kann man auch einen Webservice für die Entwicklung verwenden. Alles, was man braucht, ist ein Browser und eine Internetverbindung. Eine gute Plattform dazu ist <http://codenvy.io>.

Wenn man einen anderen Dienst verwenden möchte, ist lediglich das Wording anders, aber der Prozess ist normalerweise ziemlich ähnlich. Ein HowTo für <http://codenvy.io> stellen die RailsGirls unter <http://railsgirlshh.github.io/install> zur Verfügung.

Das HowTo ist auch als pdf-Datei auf dem Modulshare verfügbar: `/sh-modules/iet-133/03_Arbeitsblaetter/AB133_01_RailsAsCloudServiceByRailsGirl.pdf`

Wir wollen mit der installierten Version von Ruby on Rails arbeiten und führen die Installation schrittweise auf der bmLP1 durch.

Wir benutzen die Version `ruby-2.4.2@rails5.2`.

Die vollständige Release History mit Changes innerhalb der Versionen ist unter <http://railsapps.github.io/rails-release-history.html> sehr gut dokumentiert.

## Ruby on Rails auf Ubuntu

Das Betriebssystem Ubuntu ist genau wie andere unixoide Betriebssysteme (u.a. Mac OS X) sehr beliebt für die Rails Entwicklung.

Damit wir einfach zwischen den Ruby-Versionen wechseln können, ist der Einsatz des Ruby Version Manager (RVM) empfohlen. RVM bietet auch eine gute Möglichkeit, Abhängigkeiten von Gem zu verwalten.



Installieren Sie niemals Ruby als Paket mit `apt-get`. Der Paketmanager installiert Ruby als Systeminstallation, die allen Benutzern zugänglich ist. Es ist von Vorteil, wenn Sie eine eigens lokale Version installieren. Zudem sind die Ruby-Versionen, die mit dem Paketmanager installiert werden, oftmals veraltet.

## Installation von Ruby on Rails

Achten Sie darauf, dass Sie folgende Installations- und Konfigurationsschritte mit dem `vadmin`-Benutzer durchführen. Als `root` wird es nicht funktionieren.

Es kann sein, dass sich die Installationsschritte von Version zu Version ändern. Hierzu werden immer passende HowTo-Schritt in der Konsole ausgegeben. Bei Unklarheiten wenden Sie sich an die Lehrperson. Die folgende Installation basiert auf den Arbeiten von Daniel Kehoe: <http://railsapps.github.io/installrubyonrails-ubuntu.html>

Eine sehr gute Beschreibung der Installation von RVM, Ruby und Rails auf Ubuntu finden Sie im Tutorial "How to Install Ruby on Rails on Ubuntu 18.04 LTS" unter <https://www.howtoforge.com/tutorial/ubuntu-ruby-on-rails/>

Wir müssen das System für die Rails Installation vorbereiten:

1. Aktualisieren Sie das System der bmLP1, damit der Paketmanager die neuesten Versionen der Railsvoraussetzungen installiert:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

2. Installieren Sie `curl` für den RVM (Informationen zu `curl` unter <https://en.wikipedia.org/wiki/CURL>):

```
$ sudo apt-get install curl
```

3. Installieren Sie Ruby mit RVM (Ruby Version Manager) und verwenden Sie RVM, um Ruby zu installieren und Ihre Rails-Versionen zu verwalten. Wenn Sie eine ältere Version von Ruby auf Ihrem Computer installiert haben, müssen Sie diese nicht entfernen. RVM wird Ihr Ruby unberührt lassen und Ihre Shell verwenden, um alle Ruby-Aufrufe abzufangen. Alle älteren Ruby-Versionen bleiben auf Ihrem System und die RVM-Version hat in jedem Fall Vorrang.

Ruby 2.6.0 ist zum Zeitpunkt der Aktualisierung dieses Dokumentes die aktuelle Version. Sie können nach der neusten, empfohlenen Version von Ruby suchen. RVM wird die neueste stabile Ruby-Version installieren. Die einfachste Art, den RVM zu installieren, ist nachfolgend beschrieben.

Zuerst den rvm-Key importieren:

```
$ curl -sSL https://rvm.io/mpapis.asc | gpg2 --import -
```

➔ Funktioniert nicht mehr 29.01.2019, anstelle dessen folgende Schlüssel verwenden:

```
$ gpg2 --keyserver hkp://pool.sks-keyservers.net --recv-keys  
409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB
```

Die stabile rvm-Version wird wie folgt installiert:

```
$ \curl -L https://get.rvm.io | bash -s stable --ruby
```

Der Backslash vor `\curl` verhindert Konflikte. Falls `curl` als Alias deklariert wurde, wird das Alias aufgehoben.

Mit dem Flag `-ruby` ist garantiert, dass die neueste Ruby-Version installiert wird.

Um den RVM zu starten, ist folgende Eingabe notwendig:

```
source /home/vmadmin/.rvm/scripts/rvm
```

➔ Wenn das Verzeichnis `rvm` nicht vorhanden ist, ist vermutlich beim Import des rvm-Keys etwas schiefgelaufen. Gehen Sie zwei Schritte zurück und versuchen Sie den Import nochmals.

Wir wollen, dass RVM bei Systemstart automatisch gestartet wird:

```
$ cd /home/vmadmin  
$ sudo mcedit .bashrc
```

Erfassen Sie am Ende der Datei `.bashrc` folgenden Eintrag:

```
# Start RVM  
[[ -s "$HOME/.rvm/scripts/rvm" ]] && . "$HOME/.rvm/scripts/rvm"
```

Achten Sie darauf, keine Tippfehler zu machen.

4. Die JavaScript Serverumgebung `node.js` wird benötigt, damit Rails Asset Pipeline kompiliert werden können. Rails Asset Pipeline ist ein Mechanismus, um Javascript, CSS und Bilder für den Browser vorzubereiten. Zum Beispiel können Javascript und CSS eine zeitlang im Cache gelagert und so schneller wiederverwendet werden, ohne dass der Browser diese neu beziehen muss.

```
$ sudo apt-get install nodejs
```

5. Kontrollieren Sie die Version des Gem-Managers:

```
$ gem -v  
3.0.1
```

Aktualisieren Sie den Gem-Manager:

```
$ gem update --system
```

Kontrollieren Sie die Version des Gem-Managers erneut:

```
$ gem -v  
3.0.2
```

6. RVM Gemsets

Gem ist eine Software-Bibliothek oder ein Programm, mit dem das Rails Framework erweitert werden kann. Ein Gemset ist eine Vielzahl von Gems. Bei der Grundinstallation werden die Gemset global als default installiert.

Sie können Gemsets anzeigen:

```
$ rvm gemset list  
gemsets for ruby-2.6.0  
  (default)  
  global
```

Nur default und global werden bei der Installation einbezogen. Bei Fehlerausgaben muss die Konsole neu gestartet werden.

7. RVM Global Gemset

Schauen wir uns an, was im globalen Gemset installiert wurde:

```
$ rvm gemset use global  
$ gem list
```

Damit Sie fehlerfrei arbeiten können, benötigen wir die neuesten Versionen aller Gems. Schauen wir uns an, was alles veraltet ist:

```
$ gem outdated
```

Alle Gem auf der bmLP1 aktualisieren:

```
$ gem update
```

Damit die Gem-Installationen zukünftig beschleunigt werden, wollen wir die Dokumentationsinstallationen unterbinden. Dokumentationen können jederzeit im Internet nachgeschlagen werden.

Dokumentationen werden wie folgt deaktiviert:

```
$ echo "gem: --no-document" >> ~/.gemrc
```

8. Install Bundler

Das Bundler-Gem ist ein weiterer Zusatz, den wir für die Entwicklung von Rails Applikationen benötigen. Es erlaubt uns, alle verwendeten Gems einfach zu verwalten:

```
$ gem install bundler
```

## 9. Nokogiri

Nokogiri ist ein Gem, auf dem eine Vielzahl anderer Gem basieren. Nokogiri wird für jedes spezifische Betriebssystem explizit kompiliert. Die Kompilierung benötigt viel Zeit. Jedes Mal, wenn Sie das Nokogiri-Gem installieren, warten Sie bis zu fünf Minuten. Um zukünftig Zeit zu sparen, installieren wir Nokogiri in das RVM global Gemset:

```
$ gem install nokogiri
```

## 10. Ruby

Nun sind wir soweit, Ruby zu installieren. Setzen Sie die Version 2.6.0 als Standard:

```
$ rvm install ruby-2.6.0  
$ rvm default use ruby-2.6.0
```

## 11. Rails

Schliesslich sind wir bei der Installation von Rails angelangt.

Rails kann direkt in das globale Gemset installiert werden. Wir geben zuerst noch an, welche Versionen wir nutzen wollen:

```
$ rvm use ruby-2.6.0@rails5.2.2 --create
```

Installation der aktuellen, stabilen Version von Rails:

```
$ gem install rails
```

Evtl. gibt es neuere, wenig getestete Beta-Versionen von Rails. Diese könnten Sie mit der Option `-pre` installieren:

```
$ gem install rails -pre
```

➔ Ist nicht zu empfehlen, da Beta-Versionen zu wenig getestet sind!

Mit `rails -v` können Sie die installierte Version ermitteln.

# Erstellen des Workspace Ordners

Damit wir eine Rails-App erstellen können, benötigen wir einen Speicherort und einen Namen für die neue App. Rails-Entwicklerinnen und -Entwickler nutzen oftmals die Verzeichnisse `code/` oder `projects/`. Wir wollen `workspace/` verwenden.

Erstellen Sie im `/home/vmadmin` einen entsprechenden Ordner:

```
$ cd /home/vmadmin  
$ mkdir workspace  
$ cd workspace
```

# Erstellen einer neuen Rails App

Wir erstellen nun ein Projekt mit einem Gemset innerhalb des Ordners `workspace` / Ordners. Die neue Applikation nennen wir `myapp` (natürlich ist der Name frei wählbar):

```
$ mkdir myapp
```

Für das weitere Vorgehen muss man sich im Ordner `/home/vmadmin/workspace/myapp` befinden:

```
$ cd myapp
```

Ein neues Projekt mit einem spezifischen Gemset erstellen:

```
$ rvm use ruby-2.6.0@myapp --ruby-version --create
```

Die Option `--ruby-version` garantiert, dass die notwendigen Dateien im Stammverzeichnis der Applikation abgelegt werden. RVM erkennt diese Dateien und lädt die erforderliche Version von Ruby und das richtige Gemset.

Wir installieren nun das neuste Rails-Framework für die Applikation:

```
$ gem install rails
```

Schliesslich führen wir Rails aus und verwenden die Unix-Konvention mit dem Punkt, um auf das aktuelle Verzeichnis zu verweisen. Der Befehl `rails new` generiert die Standard-Rails-Starter-App.

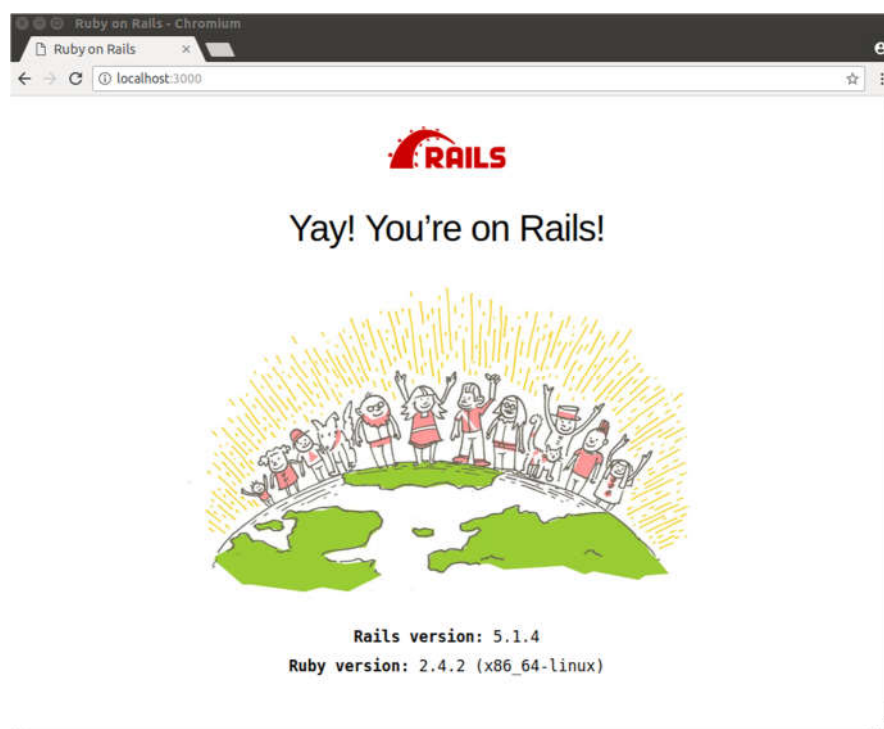
```
$ rails new .
```

Als Letztes müssen Sie noch den Rails Server starten:

```
$ rails s
```

`rails s` ist ein Alias für `rails server`

Öffnen Sie nun Google Chrome oder Firefox mit der Adresse <http://localhost:3000>







Falls anstatt der Seite "Yay! You're on Rails!" eine Fehlermeldung angezeigt wird, dass sqlite3 Version 1.3.6 nicht aktiviert werden kann, weil bereits die Version 1.4.0 aktiviert worden ist, dann tragen Sie im *Gemfile* (Verzeichnis workspace/myapp) folgendes ein:

```
gem 'sqlite3', '~> 1.3.6'
```

Stoppen Sie den Server und starten Sie ihn erneut, damit die Änderungen wirksam werden.

Damit Sie beim nächsten Systemneustart wieder auf die Seite <http://localhost:3000> gelangen können, müssen Sie den Rails-Server erneut starten:

```
$ cd /home/vmadmin/workspace/myapp  
$ rails s
```

Der Server kann mit Hilfe seiner ProzessID wieder beendet werden. Diese wird direkt in unserer App gespeichert, nämlich in der Datei  
home/vmadmin/workspace/myapp/tmp/pids/server.pid

Jeder laufende Prozess unter Linux besitzt eine solche ProzessID und kann mit dem Kommando

```
$ sudo kill <ProzessID>
```

beendet werden. Damit wir nicht immer die ProzessID des Rails-Servers für das kill-Kommando auslesen müssen, bietet sich folgende Konsoleneingabe an:

```
$ sudo kill $( cat /home/vmadmin/workspace/myapp/tmp/pids/server.pid  
)
```

Der Server muss immer nach Änderungen an Javascript und CSS Dateien neu gestartet werden. In Rails sind Javascript und CSS-Dateien Assets genannt. Statt css Dateitypen wendet Rails scss an. SCSS-Dateien können gegenüber CSS auch Schleifen und Variablen behandeln.



# Praktische Arbeit: Beim Systemstart den Rails-Server starten

Damit wir nicht ständig den Rails-Server manuell starten müssen, macht es durchaus Sinn, den Start bei einem Systemboot zu automatisieren. Es gibt mehrere Lösungsansätze unter Linux. Finden Sie eine funktionierende Lösung und testen Sie diese!

Hier einige Hinweise:

Sie wissen, dass wir den Rails-Server mit `rails s` starten.

Informieren Sie sich über die Datei `rc.local`: <https://wiki.ubuntuusers.de/rc.local/>

Lesen Sie auch, was in der Datei `rc.local` auf Ihrem System steht.

Beachten Sie insbesondere folgende Punkte:

1. Damit der Befehl `rails s` abgesetzt werden kann, müssen Sie sich im Ordner der rails app befinden (`cd /home/vmadmin/workspace/myapp`).
2. `rails s` ist kein bekanntes Linux Kommando. Es wurde bei der Installation von Rails als Skript im Ruby Version Manager (rvm) hinterlegt (`source ~/.rvm/scripts/rvm && rails s`).
3. Mit `&&` können Sie mehrere Linuxkommandos als Einheit ausführen.  
Zum Beispiel:

```
$ cd /home/vmadmin/workspace/myapp && source ~/.rvm/scripts/rvm  
&& rails s
```

4. Machen Sie mehrmals folgende Kommandozeileneingaben und probieren Sie zu verstehen was passiert:

```
$ sudo kill $( cat  
/home/vmadmin/workspace/myapp/tmp/pids/server.pid )  
$ cd /home/vmadmin/workspace/myapp && source ~/.rvm/scripts/rvm  
&& rails s
```

5. Sie sollten das Startkommando des Rails-Server für unsere App in ein Bash Skript verpacken und dieses in `rc.local` zur Ausführung bringen.

Lesen Sie hierzu: [https://wiki.ubuntuusers.de/Shell/Bash-Skripting-Guide\\_für\\_Anfänger/](https://wiki.ubuntuusers.de/Shell/Bash-Skripting-Guide_für_Anfänger/)

**Falls Sie nicht weiterkommen, wenden Sie sich an die Lehrperson.**