



Web-basierte Anwendungen

Praktikumsaufgabe 1: Statisches HTML mit HAML

Studiengänge AI (4140) & WI (4120)



Hintergrund

- **Ruby on Rails, kurz „Rails“ oder RoR**
 - RoR ist ein sehr leistungsfähiges „full stack“-Framework zur raschen Entwicklung webbasierter Anwendungen auf Serverseite
 - In dieser Eigenschaft werden wir es erst im abschließenden Projektteil verwenden, aber:
 - RoR schafft wertvolle Ordnungsstrukturen für HTML, CSS, JS und übernimmt lästige Vorverarbeitungen (HAML→HTML, SCSS→CSS, CS→JS), die auch für statische Seiten bereits nützlich sind
- **RoR in unserem Praktikum**
 - Die Praktikumsaufgaben kommen noch ohne serverseitige Code-Generierung aus.
 - Wir nutzen Rails dennoch bereits für diese eigentlich statischen Inhalte. Dadurch lernen wir Rails schrittweise kennen und können einige seiner Vorteile schon früh nutzen.
 - Benötigt: (Controller,) **Views**, bald auch CSS und JS
 - Vorläufig noch nicht relevant: Models, Datenbanken
- **WBA in diesem Semester**
 - Alle Praktikumsaufgaben sind auf Online-Lehre und Ausführung auf dem eigenen Rechner ausgelegt
 - Empfohlen: Linux (z.B. Ubuntu, auch als VM unter Windows) oder MacOS



Die Aufgabe

- Erzeugen Sie mehrere statische, miteinander verlinkte HTML-Dateien mithilfe von **HAML** und **Rails**

- Vorbereitungen:

- Wechseln Sie in ein Arbeitsverzeichnis, das Sie zur Bearbeitung der WBA-Praktikumsaufgaben eingerichtet haben, und geben Sie folgendes Kommando ein:

```
$ rails new wba1
```

Wir setzen eine halbwegs aktuelle Rails-Installation (6.x) voraus. Aktuell ist Rails 6.1.3.1 (04/2021). Zur Installation bitte die Vorübung beachten.

Es entsteht der Unterordner „wba1“. Wechseln Sie dorthin!

- Ergänzen Sie folgende Zeile in der Datei „Gemfile“:

```
gem 'haml-rails'
```

- Aktualisieren Sie Ihre installierten Projekt-Gems:

```
$ bundle install
```

- Erzeugen Sie nun Rails-Seiten („einleitung, index, kap1, kap2, tab“) zu Controller „pr01“:

```
$ rails generate controller pr01 einleitung index kap1 kap2 tab
```

Rails sollte dabei zahlreiche Dateien und Ordner erstellen, u.a. auch **Gemfile.lock**

Die Seiten entsprechen denen des Praktikums 06 aus der AI-Version von „Einführung in die Informatik“



Die Aufgabe

- Vorbereitungen (Forts.):
 - Wechseln Sie in den Ordner **app/assets** Ihres Rails-Projekts **wba1**, kopieren Sie die *.gif-Dateien aus Ordner „pr01-material“ in den Unterordner „images“
`$ cp /my/path/to/pr01-material/images/*.gif ./images`
 - Im View können Sie die Bilder später über den Pfad **/assets/*.gif** ansprechen.
Beispiel: ``
 - Wechseln Sie in Ordner **app/assets/stylesheets**, kopieren Sie die Datei **common_styles.css** aus „pr01-material“ auf Datei **pr01.scss**
`$ cp /my/path/to/pr01-material/common_styles.css ./pr01.scss`
 - Wechseln Sie in den Ordner **app/views/layouts**
 - Benennen Sie die dortige HTML-Datei um:
`$ mv application.html.erb application.html.haml`
 - **Editieren Sie diese Datei: Ändern Sie den enthaltenen HTML-Code um in HAML-Code.**
 - Hintergrund: Diese Datei bildet den Anfang aller erzeugten Views. In den anderen View-Dateien gibt man nur noch die Inhalte von „body“ an. Sie werden dann anstelle von „yield“ erscheinen!
 - Ergänzen Sie dort nun fehlende meta-Elemente aus „pr01-material/muster_50.html“
 - Tipp: Beachten Sie dabei die korrekte HAML-Syntax für Attribute



Die Aufgabe

- Vorbereitungen (Forts.):
 - Wechseln Sie nun in den Ordner **app/views/pr01**
 - Erweitern Sie die dortigen Datei-Prototypen mit den Inhalten der entsprechenden HTML-Dateien aus „wba-material“, z.B. so:

```
$ cat /my/path/to/pr01-material/einleitung.txt >> einleitung.html.haml
```
 - In den HAML-Dateien befindet sich unterhalb von etwas HAML-Platzhaltercode ein ASCII-Text mit eingebetteten Anweisungen.
- Die Aufgabe
 - Folgen Sie den Anweisungen, um die enthaltenen **Texte in HTML-Form** zu **bringen!**
 - Verwenden Sie dabei HAML als Code-Generator für HTML. Es erspart Ihnen viel Tipp-Arbeit.
 - Falls erforderlich, verwenden Sie die am Ende angegebenen Quellen, um die erforderlichen HTML- und HAML-Kenntnisse zu erwerben.
 - Für die AI-Teilnehmer(innen): Es gibt ein Werkzeug zur automatischen Umsetzung von HTML (aus Ihrer EinfInF-Aufgabe) in HAML. Widerstehen Sie der Versuchung – führen Sie diese Aufgabe manuell durch, Sie lernen dabei viel mehr!
 - Bitte beachten: Schreiben Sie in die View-Dateien nur die Inhalte von “body”! Der Grund: Alles andere generiert Rails automatisch aus Datei „views/layouts/application.html.haml”
 - Verwenden Sie für den Verweis auf den Index einfach folgenden Attribut-Inhalt:
`href="index"`. Rails ergänzt den Rest automatisch.



Die Aufgabe

- AI: Hinweise für die Verwendung von HTML-Einflnf-Quellen
 - Falls bereits eine Lösung als html-Datei vorliegt:
 - Entfernen Sie alle Zeilen außer den Inhalten von „body“
 - Entfernen Sie alle nun störenden *end tags*
 - Ersetzen Sie die *start tags* durch ihre HAML-Entsprechungen
 - Passen Sie ggf. die Attribute an
 - Beachten Sie dabei die von HAML geforderten Einrückungen
 - Verwenden Sie ggf. die HAML-Kurzformen für die Attribute *id* und *class*
 - Verwenden Sie ggf. die HAML-Kurzform für das Element *div*
- In beiden Fällen:
 - Erarbeiten Sie sich die erforderlichen HAML-Kenntnisse anhand der auf der letzten Folie angegebenen Quelle!
 - Tipp: Auf der Kommandozeile lässt sich HAML-Syntax bereits testen. Beispiel:
`$ haml -c einleitung.html.haml # c = „check“`

* Tests

- Hintergrund

- Rails enthält einen kleinen Web-Server. Im Entwickler-Modus (Normalfall) lädt er Quellcode-Änderungen automatisch, sodass Sie den Server nur einmal starten müssen. Standard-Port ist ,3000‘.
- Zum Starten (im Rails-Ordner, am besten in eigener Shell):
`$ rails server` # oder kurz: `rails s`
- Sie erreichen die Seiten (des Controllers `pr01`) nun auf dem lokalen Rechner z.B. unter `http://localhost:3000/pr01/index`

- Tests

- HAML-Fehler: Syntaxfehler werden direkt im Browser gemeldet
- HTML5-Validierungsfehler:
 - validator.w3.org in eigenem Tab aufrufen
 - Quelltext jeder Seite im Browser aufrufen und per copy/paste in die Eingabebox unter „Validate by direct input“ kopieren
 - “Check“ klicken, eventuell angezeigte Fehler beseitigen
- Die Aufgabe gilt als erfüllt, wenn alle Seiten korrekt angezeigt werden (und ggf. Hilfen für die späteren CSS-Selektionen angelegt sind)

- CSS?

- Gestaltung mit CSS erfolgt im Rahmen der nächsten Aufgabe – hier nur vorbereiten!
- Farbgebung von Kap.1: Hier nur Überschrift färben!

- **Zu CSS**

- Mit HTML bzw. HAML sollten Sie (nur) die logische Struktur einer Seite bestimmen, für ihre Gestaltung ist CSS zuständig
- CSS-Anweisungen gehören – von wenigen Ausnahmen abgesehen – nicht in die HTML-Dokumente, sondern in separate CSS-Dateien
 - In Rails finden Sie diese in **app/assets/stylesheets**
- Für das Element „div“ sowie die Attribute „id“ und „class“ – wichtig für CSS-Selektoren – gibt es in HAML Abkürzungen: Nutzen Sie sie!

- **Zu dieser Übung**

- Diese Aufgabe bildet die Grundlage für weitere Praktikumsaufgaben
- Hier geht es zunächst nur um den HTML- bzw. HAML-Code – CSS folgt im nächsten Teil
- Wer sich bereits mit CSS auskennt: Verwenden Sie Datei **app/assets/stylesheets/pr01.scss** für Ihre Experimente – ansonsten: Noch ignorieren.



Bedingungen

- Abgabe
 - Zusammen mit Aufgabe 02, in zwei Wochen
- Art des Leistungsnachweises
 - Zu vergeben: **2 Punkte**
 - **Einzel-Arbeit – keine Teams!**
 - Abgabe der Projektordner-Datei vor Beginn der dritten Übung, zusammen mit den Ergebnissen der Übung 02. Vorschau dazu:
 - Im Verzeichnis „wba1“:

```
$ rake log:clear  
$ rake tmp:clear  
$ cd ..  
# Ordner ./wba1 verpacken, ohne unnötige Unterordner:  
$ tar czf 02-wba1-<matnr>.tar.gz --exclude wba1/tmp \  
--exclude wba1/node_modules --exclude wba1/log ./wba1
```
 - Datei „02-wba1-**<matnr>**.tar.gz“ abgeben (in Stud.IP hochladen)
 - Abnahmegespräch / Online-Demo der korrekten Funktion
 - Online, per Breakout-Session.

Fortsetzungszeile

- **Zu HTML, aber auch zu XML und CSS:**
 - **de.selfhtml.org**
 - Früher der Standard
 - Immer noch gut, aber leider teilweise auf Stand 2007 stehengeblieben
 - **www.w3schools.com**
 - umfassend, gut, aktuell
 - auf Englisch
 - **www.w3.org**
 - Ort der meisten Standard-Dokumente des WWW, insb. zu HTML5 und CSS
- **Zu HAML**
 - **haml.info**
 - „Tutorial“ und „Documentation“ genügen völlig
 - **<http://www.heise.de/ix/artikel/Licht-in-Sicht-506987.html>**
 - Einführender Artikel, der sogar viele „Tricks“ beschreibt
- **Zu Rails**
 - **rubyonrails.org**
 - Kap. 1 – 3 von: https://guides.rubyonrails.org/getting_started.html