# Ein Bild, das Screenshot enthält. Mit hoher Zuverlässigkeit generierte BeschreibungQuicknotes M151

Winkler Olivier



Ein Bild, das Text, Karte enthält.

Mit sehr hoher Zuverlässigkeit generierte Beschreibung

# Quicknote AB151-01



## Active Record

* M in MVC
* Model speichert alle Daten und erleichtert Erstellung und Verwendung von Business Objects
* Darstellung von Modellen und Daten, Beziehungen zwischen Modellen, Vererbungshierarchie durch verwandte Modelle
* Zusätzlich ist es ein Entwurfsmuster (Software Pattern) 🡪 Datenobjekt beschreibt.
* Zelle in Tabelle = Objekt
* Status von Objekt kann überprüft werden bevor es in DB geschrieben wird

## ORM (Object-realational mapping)

* Ist eine Technik die Objekte einer Anwendung mit Tabellen in einem RDBMS zu verbinden.
* In einem ORM können Eigenschaften und Beziehungen der Oebjekte in einer Andwenung gespeichert und abgerufen werden ohne grosse SQL-Abfraegen.

## Aufgabe 1

*Beantworten Sie folgende Fragen unter Berücksichtigung der Namenskonvention:*

1. *Was bedeutet 2.4.2 :001 > Book.connection ?*
2. *Was bedeutet 2.4.2 :001 > Book ?*
3. *Was bedeutet 2.4.2 :001 > Book.all ?*
4. Verbindung zu Model «Book»
5. Model von Book wird angezeigt (Attribute mit Datentypen)
6. Alle Datensätze werden ausgegeben

## Konventionen

Active Record verwendet standardmässig einige Namenskonventionen. In Rails werden Tabellen pluralisiert. Somit wird für die Klasse «Book» eine Datenbanktabelle namens «books» angewendet. Bei Rubyklassen werden die Erstbuchstaben aller Wörter grossgeschrieben. Modellklassennamen werden in CamelCase geschrieben.

Klasse 🡪 BuchHandel

Tabelle 🡪 buch\_handels

## Aufgabe 2

*Vervollständigen Sie folgende Tabelle. Beachten Sie das alle Modell- und Klassennamen Englisch geschrieben sind.*

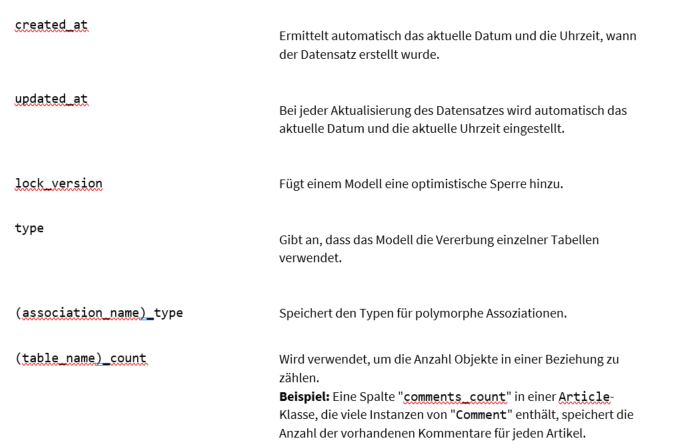


## Schema Konventionen

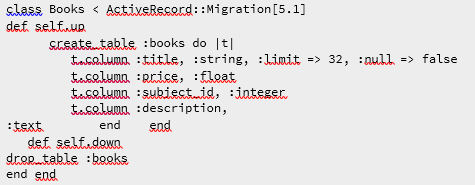
Active Record verwendet unterschiedliche Namenskonventionen für Attribute in Datenbanktabellen z.B. Fremdschlüssel.

singularized\_table\_name\_id (z.B. book\_id, subject\_id)

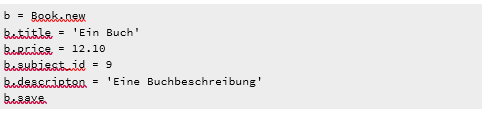
Active Record erkennt diese Schreibweise als Beziehung zwischen Tabellen.

Genauso verwendet Active Record standardmässig eine ganzzahlige Spalte mit dem Namen «Id» und dient als Primärschlüssel der Tabelle. Bei einer Migration wird diese Spalte automatisch erstellt und auch folgende Attribute:

## Active Record Modell erzeugen



## Objekt in Modell



## Aufgabe 3

*Erstellen Sie ein Objekt Book mit der folgendem Active Record entspricht:*

*#<Book title: "Linux Server", price: 13.0, subject\_id: 2, description: "Linux is faster"> Nutzen Sie die create-Methode:*

Book.create(title: “Linux Server”, price: 13.0, subject\_id: 2, description: “Linux is faster”)

*Nutzen Sie für den Active Record*

*#<Book title: "Ruby Book", price: 21.00, subject\_id: 8, description: "Simple as that"> die new-Methode sowie save-Methode:*

B = Book.new

B.title = “Ruby Book”

B.price = 21.00

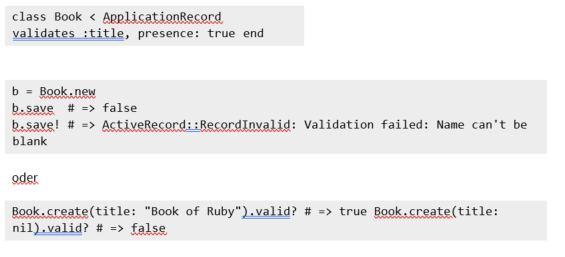
B. subject\_id: 8

B.description = “Simple as that”

B.save

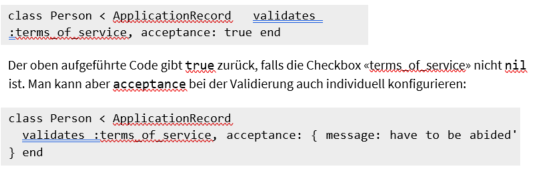
## Validierung

Mit Active Record kann der Status eines Objekts überprüft werden, bevor es in die Datenbank übertragen wird. Geprüft wird z.B. ob ein Feld leer ist oder den richtigen Wert hat. Die Validierung wird bei der Speicherung in eine Datenbank überprüft. Die Validierungsmethoden geben im Fehlerfall ein «false» zurück.



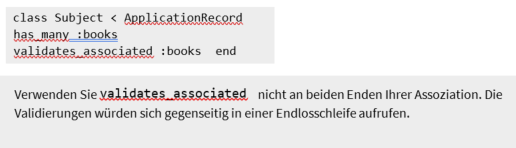
## Acceptance

Diese Methode wird für die Validierung einer Checkbox verwendet, ob diese aktiv oder inaktiv ist.



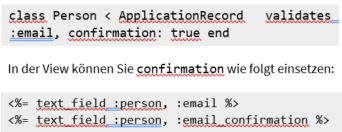
## Validates\_associated

Diese Methode wird für die Validierung einer Beziehung verwendet. Bei der Speicherung eines Objekts wird «valid?» auf die zugehörigen Objekte angewendet.



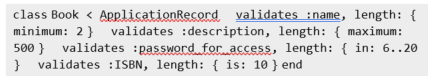
## Confirmation

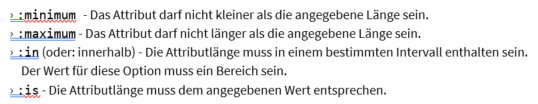
Diese Methode wird verwendet, wenn zwei gleiche Validierungen auf zwei Textfelder zutreffen sollen z.B. Bestätigung Passwort oder Email.



## Length

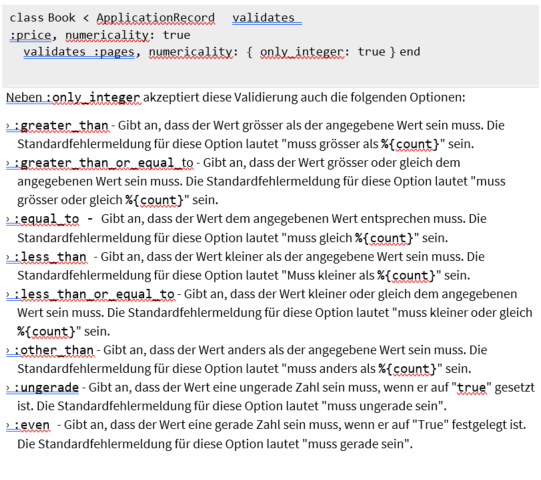
Diese Methode kontrolliert die Länge von Attributen.





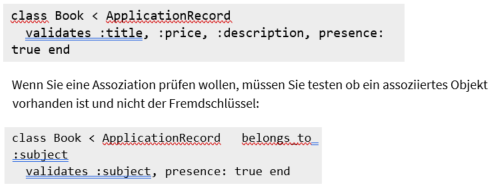
## Numericality

Diese Methode überprüft ob der Wert nur numerisch ist. Dabei werden die Werte auf die Datentypen «Integer» und «float» geprüft.

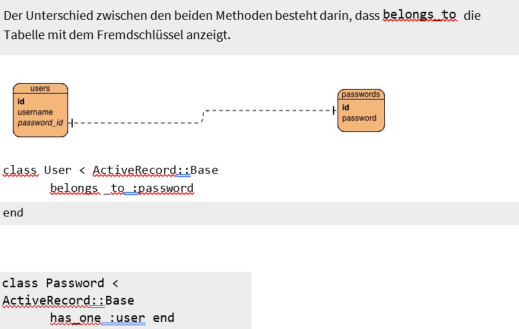


## Presence

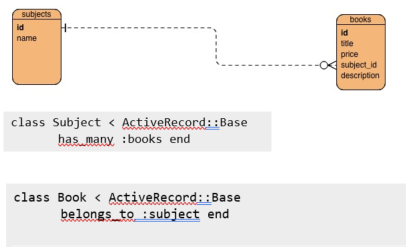
Diese Validierung prüft, ob der angegebene Wert nicht leer ist.



## Modell Assoziationen

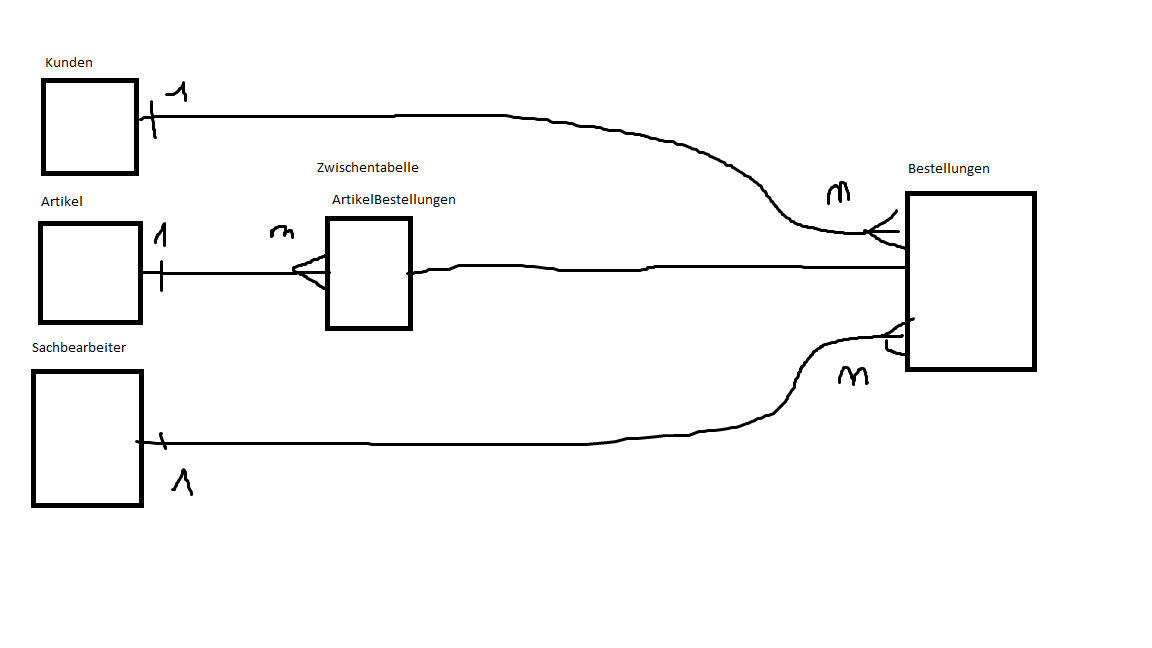
1:1 Beziehungen

1:n Beziehungen



N:n Beziehungen

Beziehung von mehreren Tabellen



**Was habe ich gelernt?**

Ich lernte in diesem Arbeitsblatt sehr viel über Active Record. Im Vormodul habe ich noch nichts mit dem Active Record gemacht. Ich lernte die Grundlagen zum Active Record und baute ihn in meine Webapplikation ein. Der Active Record dient zur Benutzerverwaltung auf meiner Webseite.

**Was hat mich behindert?**

Am Anfang hatte ich etwas Mühe meine Applikation zum Laufen zu bringen. Ich hatte immer wieder kleinere Fehler beim Aufsetzten und ich konnte diese anschliessend beheben, so dass meine Applikation nun reibungslos funktioniert.

**Was habe ich nicht verstanden?**

Am Anfang hatte ich noch etwas Mühe alles zusammen zu verstehen. Als ich es dann aber in Ruhe durchgelesen und angewendet habe, ist mir alles klargeworden.

**Was kann ich beim Studium besser machen?**

Ich bin sehr zufrieden mit mir. Ich finde momentan kann ich mich nicht gross verbessern, da ich finde, dass ich sehr gut arbeite und gut im Zeitplan bin.

Habe ich das erreicht, was ich wollte? Bin ich mit mir zufrieden?

100%

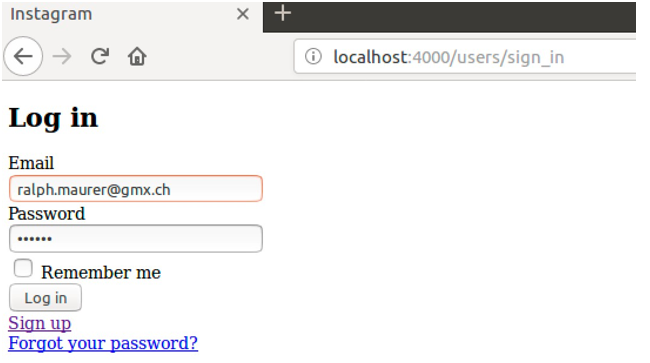
90%

# Quicknote AB151-02



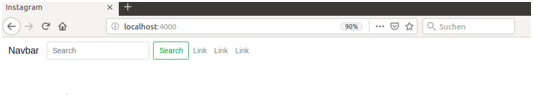
## Login

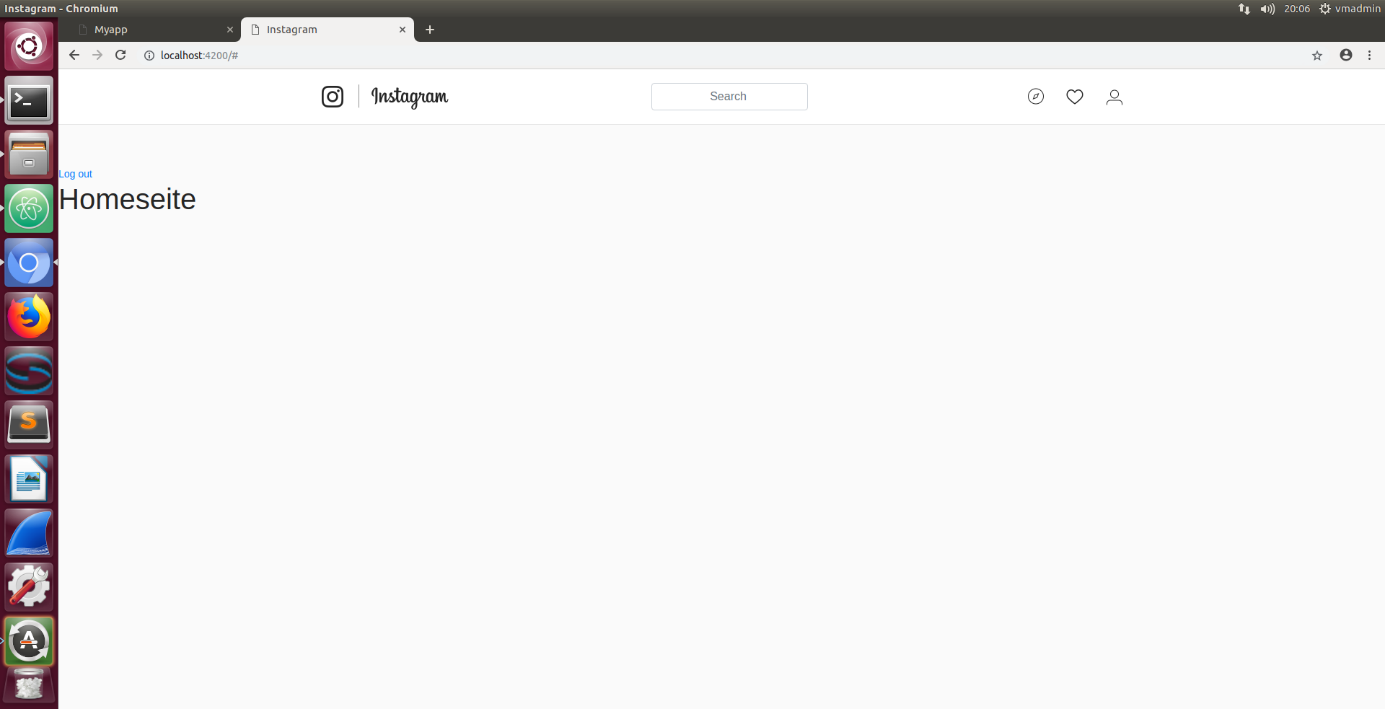
In diesem Arbeitsblatt habe ich angefangen Instagram nachzubauen. Zuerst habe ich eine Authenifizierung für meine App eingebunden. Diese habe ich nicht selbst gemacht, sondern mit einem gem (Devise) installiert. Mit diesem gem wird ein login mitsamt Registration generiert. Zusätzliche verwende ich Bootstrap für dieses Projekt. Bootstrap kann überall verwendet werden. Es ist ein Framework, welches einen schnellen Start bei der Programmierung einer Webapp ermöglicht.

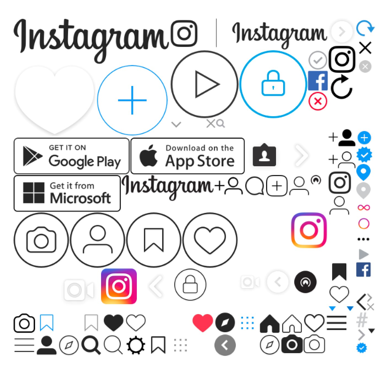


## Header

Als nächstes habe ich angefangen den Header von Instragram nachzubauen. Dies habe ich mit einer Navbar gemacht, welche aus Bootstrap stammt. Mit verschiedenen CSS-Klassen habe ich die Navbar so gestaltet, dass sie fast wie im Original aussieht. Die verschiedenen Bilder auf der Seite sind keine einzelnen, sondern das Bild hat alle Icons abgebildet, die benötigt werden und wird dann einfach zugeschnitten auf das benötigte Icon.

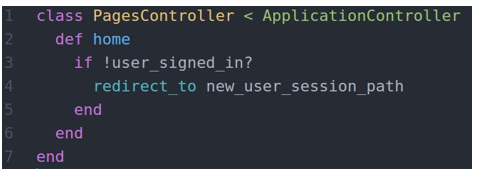






## Logout

Zum Schluss habe ich noch ein Logout implementiert. Falls der User noch von der letzten Session eingeloggt ist, wird er abgemeldet.



**Was habe ich gelernt?**

In diesem Arbeitsblatt lernte ich viel über Bootstrap. Im Vormodul haben wir auch schon bereits mit Bootstrap gearbeitet aber noch nicht allzu viel mit den Bootstrapklassen gearbeitet. Zudem habe ich gelernt wie man ein Bild in SCSS zuschneidet.

**Was hat mich behindert?**

Bis Bootstrap ohne Probleme lief, hatte ich einige kleine Probleme welche ich aber mit der Zeit lösen konnte.

**Was habe ich nicht verstanden?**

Ich habe alles verstanden und habe keine Unklarheiten

**Was kann ich beim Studium besser machen?**

Ich bin immer noch sehr zufrieden wie ich vorwärts komme und denke, dass ich weiter hin so weiterfahre.

Habe ich das erreicht, was ich wollte? Bin ich mit mir zufrieden?

100%

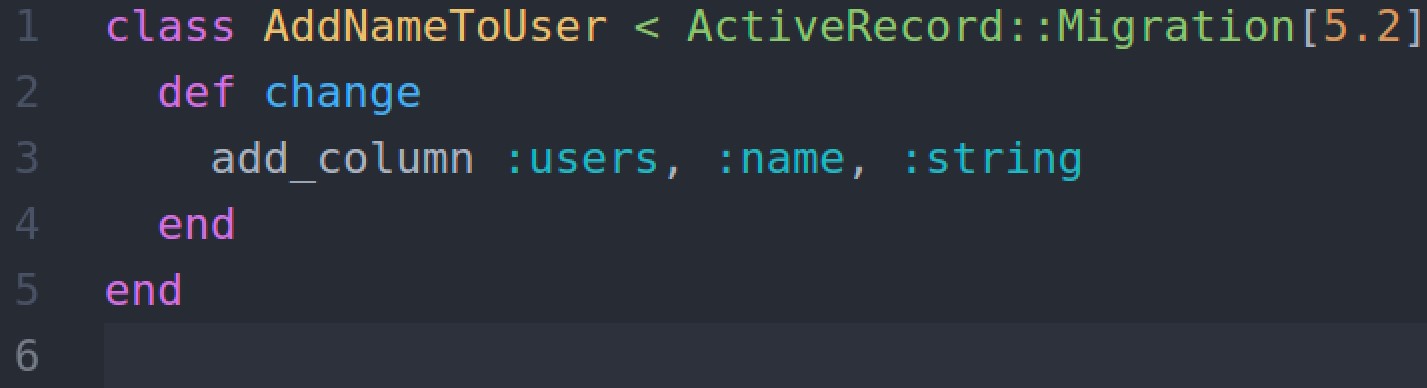
100%

# Quicknote AB151-03



## Login & SignUp

Mit dem dritten Arbeitsblatt habe ich den Login- & SignUpScreen nach Instagram erstellt. Zuerst musste ich eine neue Migration erstellen. In dieser Migration erfasste ich ein «Namen»-Attribut.



## Validierung Benutzernamen

Ein Bild, das Screenshot, Bildschirm, drinnen enthält.

Mit hoher Zuverlässigkeit generierte BeschreibungDanach musste ich eine Validierung für diesen Benutzernamen erstellen. Die Validierung prüft ob der überhaupt ein Name / Buchstaben eingegeben wurden und ob dies nicht über 50 Zeichen in der Länge misst.

## Navbar ausblenden, wenn Benutzer nicht eingeloggt

Ein Bild, das Screenshot, Monitor, Bildschirm enthält.

Mit sehr hoher Zuverlässigkeit generierte BeschreibungDer nächste Schritt war die Navbar am oberen Bildschirm auszublenden, wenn der Benutzer nicht eingeloggt ist.

Danach habe ich das IPhone-Bild auf der Startseite eingefügt.

Ein Bild, das Screenshot, drinnen enthält.

Mit sehr hoher Zuverlässigkeit generierte Beschreibung

Ein Bild, das Screenshot, Bildschirm, Monitor enthält.

Mit hoher Zuverlässigkeit generierte Beschreibung

## SCSS

SCSS ist das aufgeräumte CSS. SCSS hat die gleiche Syntax wie CSS. SCSS hat den Vorteil, dass man die verschiedenen Kinderelemente / Klassen verschachteln kann. So ist es auch übersichtlicher und spart Platz, da man für jede Klasse diese nur einmal in ein SCSS-File schreiben muss anstatt wie beim gewohnten CSS.

CSS Syntax

.klasse1 .klasse2{

Color: black;

}

SCSS Syntax

.klasse1{

.klasse2{

Color: black;

}

}

Ein Bild, das Screenshot enthält.

Mit sehr hoher Zuverlässigkeit generierte Beschreibung

## Loginscreen

Als nächstes designte ich das Loginfield.

Ein Bild, das Screenshot, drinnen enthält.

Mit sehr hoher Zuverlässigkeit generierte Beschreibung

Ein Bild, das Screenshot, Monitor enthält.

Mit sehr hoher Zuverlässigkeit generierte Beschreibung

## Footer

Als nächstes erstellte ich einen Footer mit Links zu Support etc. Ein Bild, das Screenshot, drinnen enthält.

Mit sehr hoher Zuverlässigkeit generierte Beschreibung

Ein Bild, das Screenshot enthält.

Mit sehr hoher Zuverlässigkeit generierte Beschreibung

## Signupscreen

Ein Bild, das Screenshot enthält.

Mit sehr hoher Zuverlässigkeit generierte BeschreibungDanach musste ich die Registrationsview gestalten. Die Registrationsview hat das gleiche Layout wie bei der Loginseite, nur muss man in dem Anmeldefenster zusätzlich noch seinen Benutzernamen festlegen und das Passwort zur Bestätigung nochmals eingeben.

Ein Bild, das Screenshot, drinnen enthält.

Mit sehr hoher Zuverlässigkeit generierte Beschreibung

**Was habe ich gelernt?**

In diesem Arbeitsblatt lernte ich nicht allzu viel. Das Meiste habe ich schon früher angeschaut. Ich konnte aber viel repetieren wie SCSS oder auch Bootstrapklassen.

**Was hat mich behindert?**

Manchmal hatte ich Mühe zu verstehen was gemeint ist auf den Arbeitsblättern. Nach einigen Malen durchlesen habe ich es aber immer verstanden.

**Was habe ich nicht verstanden?**

Ich habe alles verstanden und habe keine Unklarheiten

**Was kann ich beim Studium besser machen?**

Ich bin immer noch sehr zufrieden wie ich vorwärtskomme und denke, dass ich weiter hin so weiterfahre.

Habe ich das erreicht, was ich wollte? Bin ich mit mir zufrieden?

95%

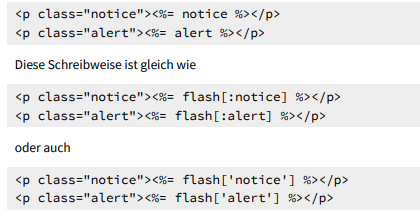
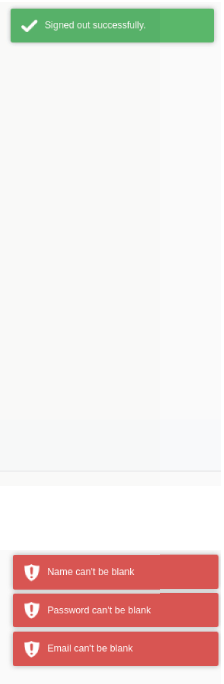
95%

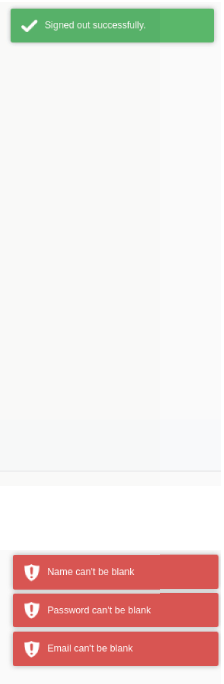
# Quicknote AB151-04



## Flash Notifactions in Rails

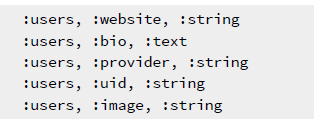
Als Erstes habe ich in diesem Arbeitsblatt Flash Notifications eingebunden. Diese geben dem Benutzer ein Feedback bei richtigen oder falschen Angaben.

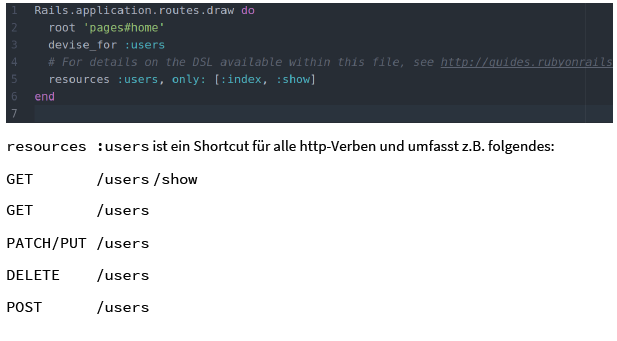




## Erweiterung Model *User*

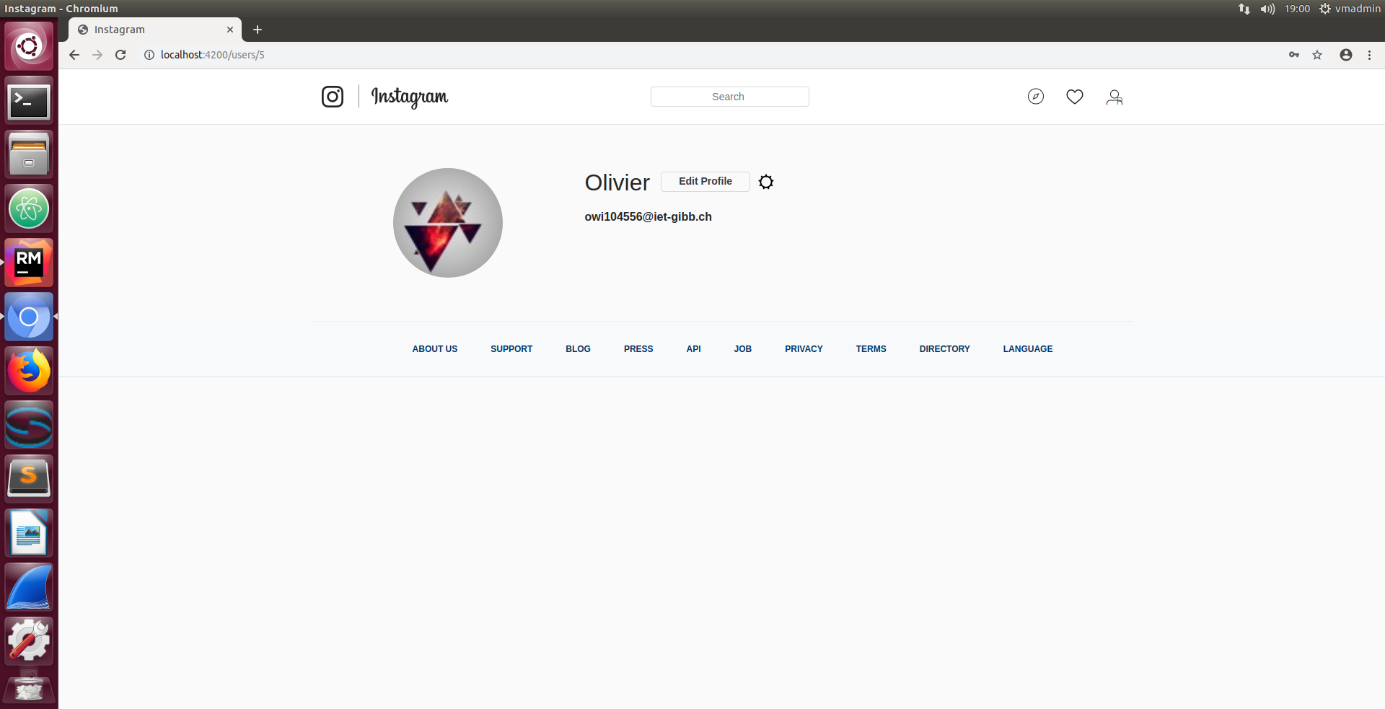
Ich habe ein neues Model zur Tabelle User hinzugefügt. Damit kann ich später jedem Profil ein Profilbild geben. Zudem kommen noch weitere Felder, die der Benutzer dann in seinem Profil angeben kann z.B. eine Webseite in seiner Profilbeschreibung. Zudem habe ich die Routen aktualisiert für die Tabelle User

Neue Migration *AddFieldsToUser* in der Tabelle *users* mit folgenden Attribute



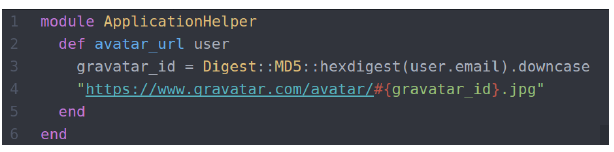
Mit den Einschänkungen [: index, :show] werden die Methoden auf die Views index & show in app/views/users/\*.html.erb gebunden

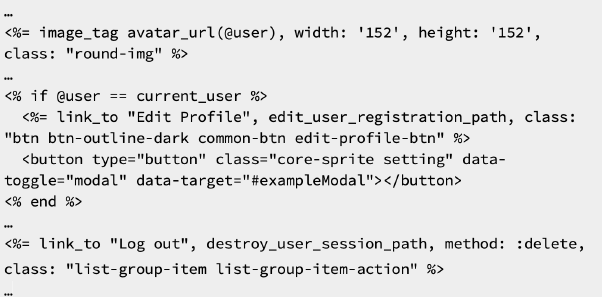
## Gravatar

Mit Gravatar haben wir Profilbilder für bestimmte Emailadressen verknüpft. So hat jeder Account mit einer registrierten Mail auf Gravatar das dort festgelegte Icon als Profilbild.

**Einbindung in Instragram**

Das Profilbild wird über ein Modul in die Applikation eingebunden. Um das richtige Bild von dem Server zu holen, wird die ID im Modul mit der Email ausgelesen und anschliessend wird diese ID an die URL angehängt und so wird das spezifische Bild an das Profil angebunden.



Der untenstehende Code zeigt den «Edit Profile» Button bei dem Profil an, welches angemeldet ist.

**Wie & wo können die vorgestellten Techniken, Methoden & Konzepte in einer Rails-App angewendet werden?**

Die Notifications können für fast jede Applikation, die mit dem Benutzer interagiert, verwendet werden um ihn z.B. auf seine Fehleingaben hinzuweisen.

**Was sind Vor- und Nachteile?**

Das Plugin kann bei instabiler Internetverbindung zu kleinen Problemen führen.

**Was habe ich gelernt?**

Ich habe gelernt wie die Flashnotifactions in Ruby funktionieren. Zudem habe ich gelernt wie man ein Profil via Email in eine Applikation integriert.

**Was hat mich behindert?**

Ich hatte einige Probleme mit Gravatar bezüglich der Verbindung. Das Problem liegt an dem Internet in dem Klassenzimmer, da dies die ganze Zeit Unterbrüche verzeichnet.

**Was habe ich nicht verstanden?**

Ich habe alles verstanden und habe keine Unklarheiten

**Was kann ich beim Studium besser machen?**

Ich bin immer noch sehr zufrieden wie ich vorwärtskomme und denke, dass ich weiter hin so weiterfahre.

Habe ich das erreicht, was ich wollte? Bin ich mit mir zufrieden?

95%

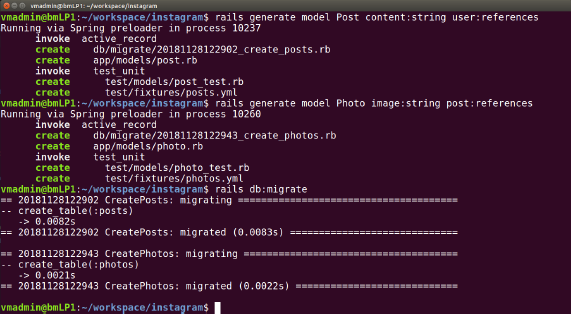
90%

# Quicknote AB151-05



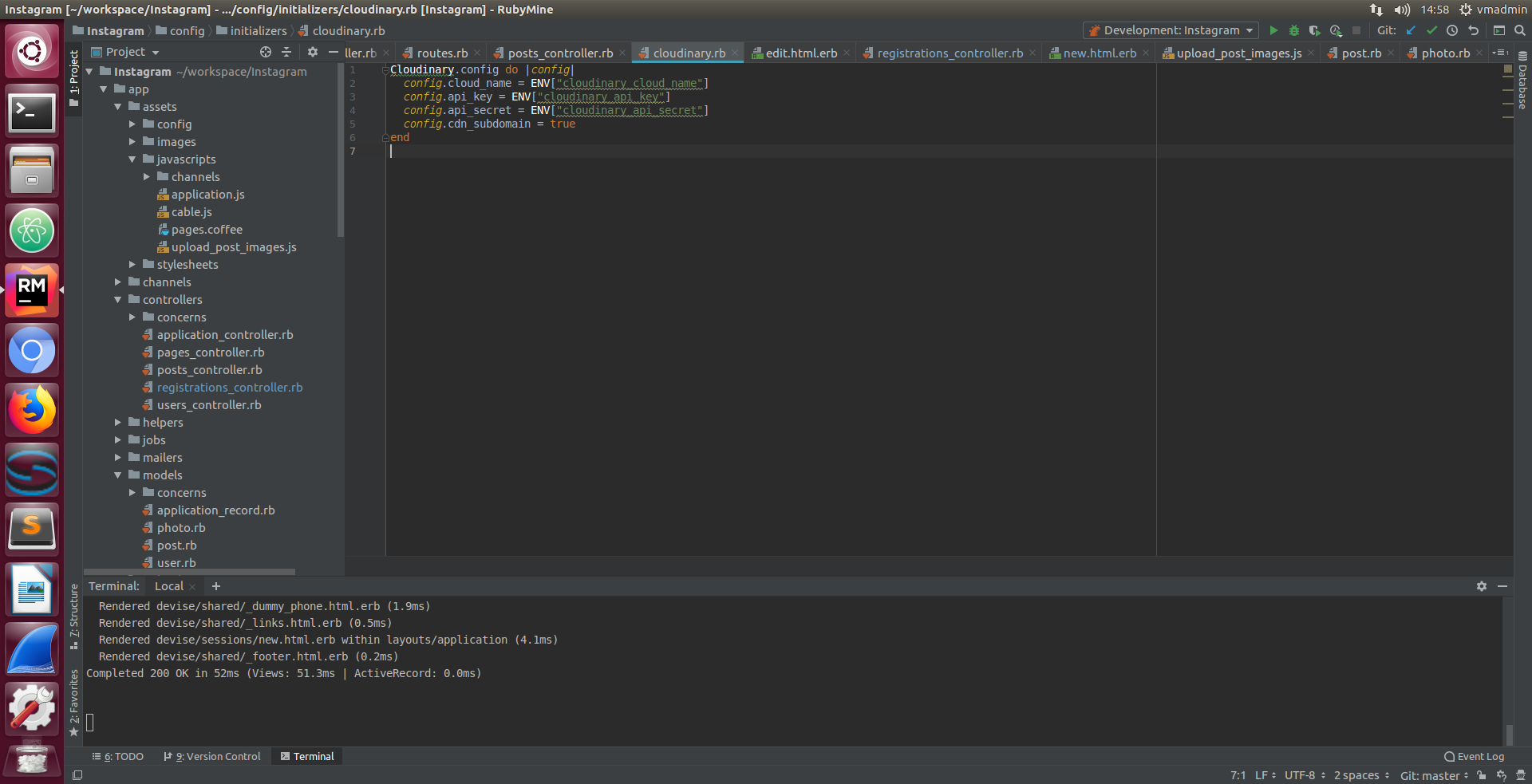
## Model von Post & Photo verbunden mit Tabelle User

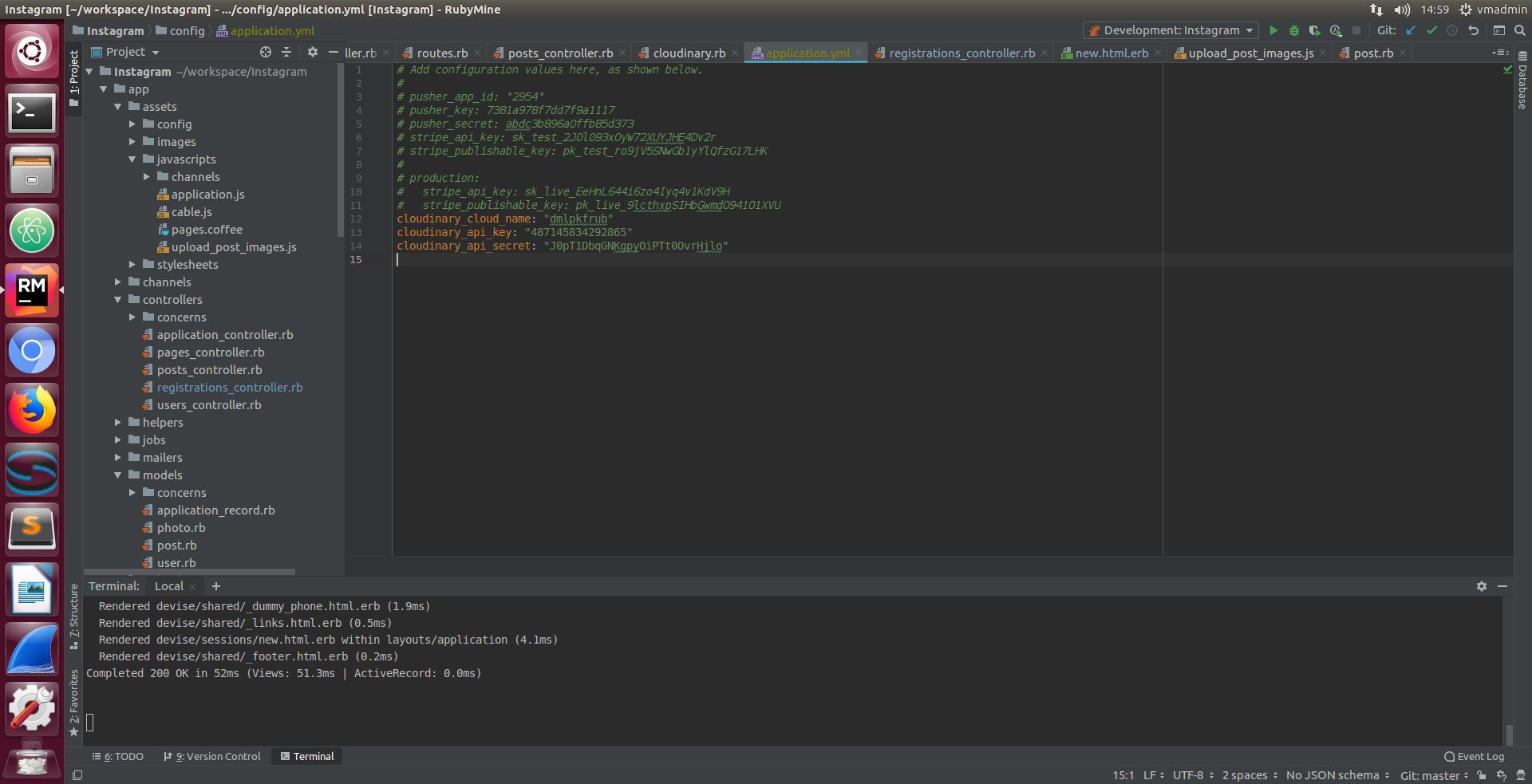
In diesem Arbeitsblatt ging es um die Posts. Zuerst musste ich dafür ein neues Model für die Posts und Photos erstellen.



## Cloudinary

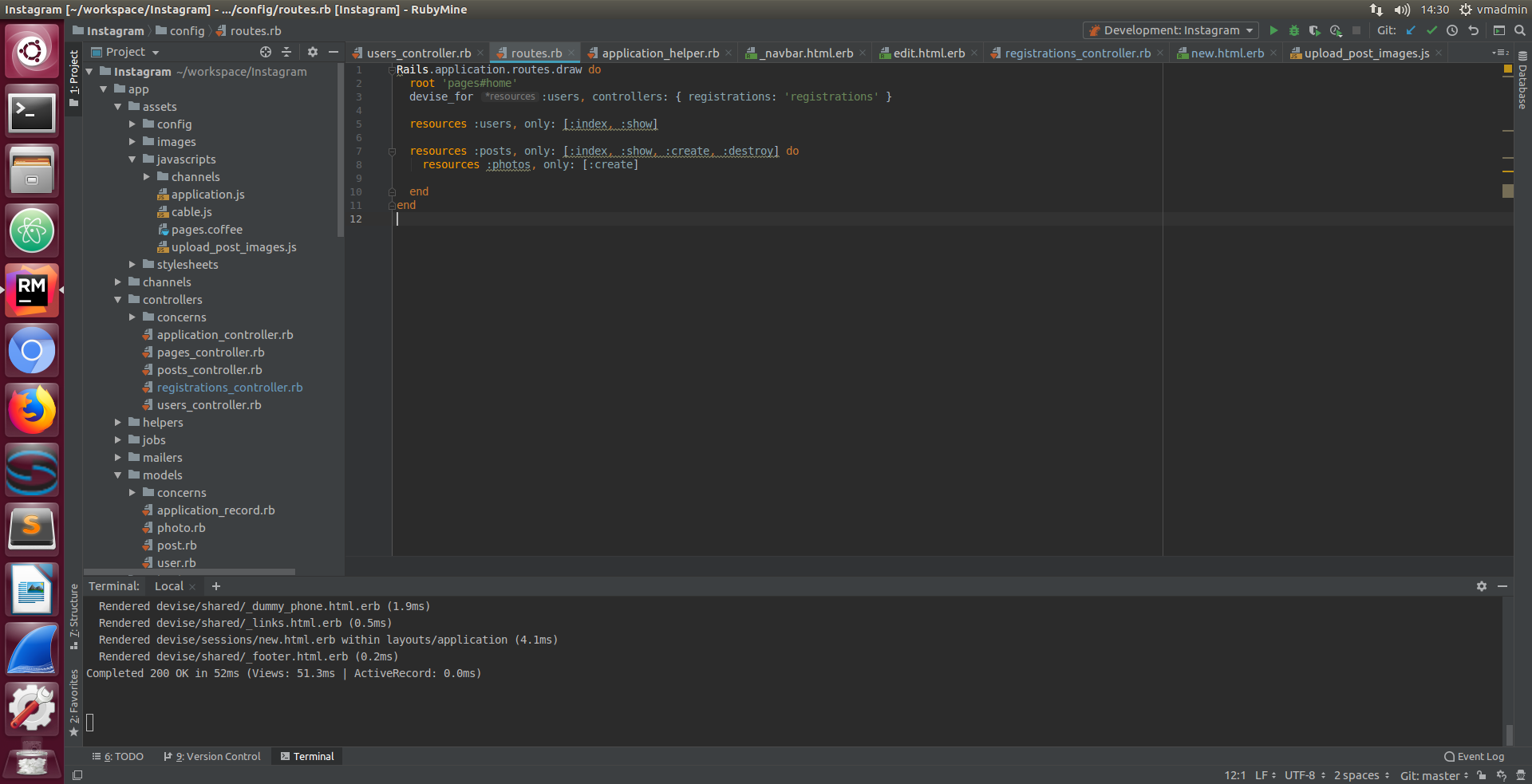
Ich habe in meine Applikation Cloudinary eingebunden, welches ermöglicht Fotos zu speichern und diese dann anzuzeigen. Dabei werden einige Systemvariablen gesetzt. Diese werden dann noch mit einem Plugin verschlüsselt, dass diese persönlichen Daten von den Variablen nicht auf dem Repo und gegen aussen sichtbar sind.





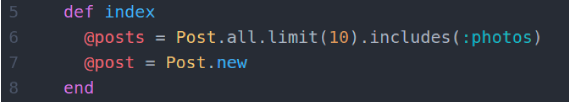
## PostController

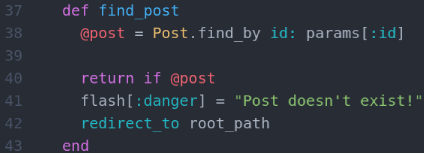
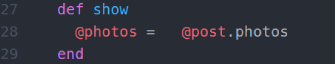
Um die Posts anzuzeigen zu können und zu verwalten habe ich noch einen Controller erstellt. Dieser hat alle untenstehenden Methoden. Zudem musste ich die Routen für die Posts anpassen.

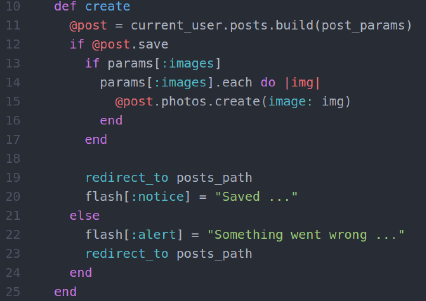
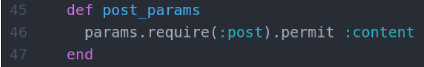
Methoden in den Routes und Controllern

Mit den beiden Zeilen sagen wir, dass nur angemeldete Benutzer darauf zugreifen können.



In der Indexmethode werden die neusten 10 Post zurückgegeben und es kann ein neuer Post erfasst werden.

Diese Methoden finden einen spezifischen Post und zeigen die Posts an.

Diese beiden Methoden sind dafür da einen Post zu erstellen.

In der showview wird dann in einer Schlaufe jeder einzelne Post ausgelesen und angezeigt.



**Wie & wo können die vorgestellten Techniken, Methoden & Konzepte in einer Rails-App angewendet werden?**

Der PostController ist für generelle CRUD-Applikationen geeignet. Das Plugin für das Hochladen von Fotos.

**Was sind Vor- und Nachteile?**

Das Plugin funktioniert nur bei einer durchgehenden Internetverbindung.

**Was habe ich gelernt?**

Ich habe gelernt wie man ein Plugin um ein Foto hochzuladen, installiert und in die Applikation einbindet.

**Was hat mich behindert?**

Die Internetverbindung im Schulzimmer.

**Was habe ich nicht verstanden?**

Ich habe alles verstanden und habe keine Unklarheiten

**Was kann ich beim Studium besser machen?**

Ich bin immer noch sehr zufrieden wie ich vorwärtskomme und denke, dass ich weiter hin so weiterfahre.

Habe ich das erreicht, was ich wollte? Bin ich mit mir zufrieden?

95%

90%