



TIM

aka EPROG V2.0

Martin Schürerer

Ruby

Examples

Rails

Examples
Why Rails?

TIM
aka EPROG 2.0

What we did
When to choose Rails

Ruby

Ruby

Dynamic typing

Fully object oriented

Metaprogramming

Interpreted

```
5.times do  
  puts "jsug".upcase  
end
```

JSUG
JSUG
JSUG
JSUG
JSUG

```
unis = ["TU", "Uni", "WU"]  
unis = unis.map do |o|  
  o + " Wien"  
end
```

```
=> unis = ["TU Wien", "Uni Wien", "WU Wien"]
```

```
unis = ["TU", "Uni", "WU"]  
unis.map! do |o| "#{o} Wien" end
```

```
=> unis = ["TU Wien", "Uni Wien", "WU Wien"]
```

```
unis = ["TU", "Uni", "WU"]  
unis.map! { |o| "#{o} Wien" }
```

```
=> unis = ["TU Wien", "Uni Wien", "WU Wien"]
```



```
options = {:auto_indent => true, :shift_width => 2}  
puts " " * options[:shift_width] if options[:auto_indent]
```

```
class Water < Object
  def initialize
    @color = "aquamarine"
  end

  def color
    @color
  end
end
```

```
class Water < Object
  attr_reader :color

  def initialize
    @color = "aquamarine"
  end
end
```

```
class Water < Object
  attr_reader :color

  def initialize
    @color = "aquamarine"
  end
end

w = Water.new
puts w.color
```

„When I see a bird that
walks like a duck and
swims like a duck and
quacks like a duck,
I call that bird a duck.“



```
class Duck
  def quack
    puts "Quack!"
  end
```

```
  def feed(food, caregiver)
    quack
    bite(food)
  end
end
```

```
class Swan
  def quack
    puts "Tzzzz"
  end
```

```
  def feed(food, caregiver)
    bite(food)
    bite(caregiver) if rand() > 0.8
  end
end
```

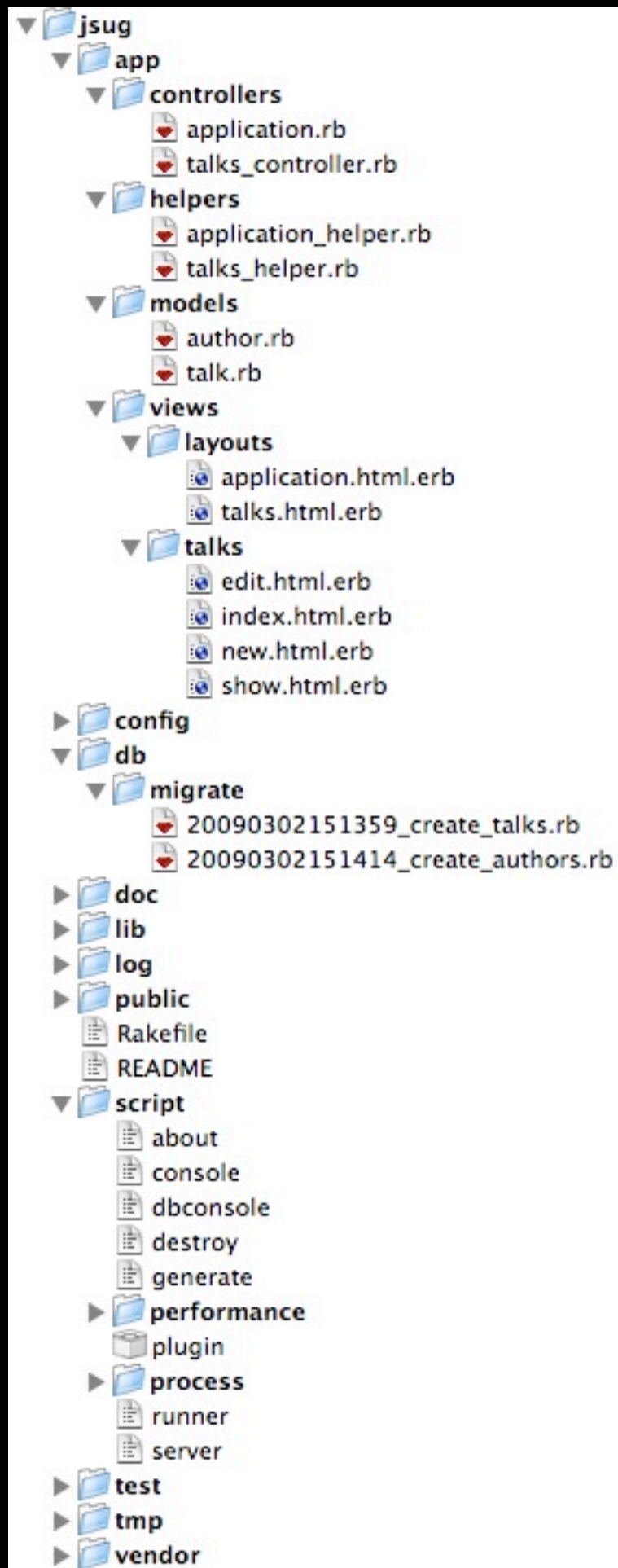
```
zoo_animals = [Duck.new, Swan.new, Duck.new, Lion.new]
zookeeper = Zookeeper.new
```

```
zoo_animals.each do |animal|
  if(animal.respond_to? :quack)
    animal.feed(Bread.new, zookeeper)
  end
end
```

```
File.open "todolist.txt", "w" do |f|  
  f.puts "- Ruby lernen"  
  f.puts "- Haskell lernen"  
  f.puts "- Scala lernen"  
end
```

```
File.open "todolist.txt", "r" do |f|  
  f.each do |line|  
    puts line  
  end  
end
```

Ruby on Rails





Welcome aboard

You're riding Ruby on Rails!

[About your application's environment](#)

Getting started

Here's how to get rolling:

1. Use `script/generate` to create your models and controllers

To see all available options, run it without parameters.

2. Set up a default route and remove or rename this file

Routes are set up in `config/routes.rb`.

3. Create your database

Run `rake db:migrate` to create your database. If you're not using SQLite (the default), edit `config/database.yml` with your username and password.

Search the Rails site

Join the community

[Ruby on Rails](#)

[Official weblog](#)

[Wiki](#)

Browse the documentation

[Rails API](#)

[Ruby standard library](#)

[Ruby core](#)

```
class Author < ActiveRecord::Base
  has_many :authorships
  has_many :books, :through => :authorships
end
```

```
class Book < ActiveRecord::Base
  has_many :authorships
  has_many :authors, :through => :authorships
end
```

```
class Authorship < ActiveRecord::Base
  belongs_to :author
  belongs_to :book
end
```

```
david, awrd = Author.find(1), Book.find(1)
david.authorships.create(
  :book => awrd, :contribution => "partial", :written_in => "Denmark"
)
```

```
david.authorships.first.written_in # => "Denmark"
david.books.first # => awrd
```

```
class Story < ActiveRecord::Base
  belongs_to :iteration
  acts_as_list :scope => :iteration
end
```

```
story.move_higher
story.move_to_bottom
```

```
<h1>Create a new iteration</h1>
```

```
<p>
```

```
  You already have <%= pluralize(@iterations.size, "iteration") %>.
```

```
  The last one started like this:
```

```
  "<%= truncate(@iterations.first.description, 10) %>"
```

```
</p>
```

```
<%= form_for :iteration, Iteration.new do |i| %>
```

```
  Title:<br/>
```

```
  <%= i.text_field :title %>
```

```
  Description:<br/>
```

```
  <%= i.text_area :title, :size => "20x40" %>
```

```
  Completion target:<br/>
```

```
  <%= i.date_select :completed_on, :discard_year => true %>
```

```
<%= fields_for :story, Story.new do |s| %>
```

```
  Headline:<br/>
```

```
  <%= s.text_field :headline, :length => 30, :class => "big" %>
```

```
  Difficulty:<br/>
```

```
  <%= s.select :difficulty, %w( easy medium hard) %>
```

```
<%= end %>
```

```
<%= end %>
```

```
...or <%= link_to "Cancel", :action => "overview" %>
```

New talk

Title

Author

Starts at

 — :

Ends at

 — :

[Back](#)

Listing talks

Title	Author	Starts at	Ends at	
TIM - Ruby on Rails 0		2009-03-02 20:24:00 UTC	2009-03-02 21:00:00 UTC	Show Edit Destroy

[New talk](#)

```
class Account < ActiveRecord::Base
  validates_presence_of    :subdomain, :name, :email_address, :password
  validates_uniqueness_of  :subdomain
  validates_acceptance_of  :terms_of_service, :on => :create
  validates_confirmation_of :password, :email_address, :on => :create
end
```



```
class StoryController < ApplicationController
  session      :off,          :only => :feed
  verify       :method => :post, :only => [ :comment ]
  before_filter :authenticate, :only => [ :new, :edit ]

  def show
    @story = Story.find(params[:id])
  end

  def new
    if request.story?
      @story = Story.create(params[:story])
      cookies[:last_story] = {
        :value    => @story.created_at,
        :expires  => 20.years.from_now
      }

      flash[:notice] = "Created new story"
      redirect_to :action => "show", :id => @story
    else
      @story = Story.new
      render :layout => "creation"
    end
  end
end
```

```
ActionController::Routing::Routes.draw do |map|
  map.start    '', :controller => 'pages', :action => 'start'
  map.signup   'signup', :controller => 'account', :action => 'new'
  map.page     'page/:id', :controller => 'pages', :action => 'show'
  map.connect  'page/:id/:action', :controller => 'pages'
  map.connect  'images/icons/:icon_file', :controller => "assets", :action => "missing_icon"

  map.feed     'feed/:token', :controller => "account", :action => "feed"
  map.ical     'ical/:token', :controller => "reminders", :action => "ical"
end
```

```
class Person < ActiveRecord::Base
  acts_as_taggable
end
```

```
class Message < ActiveRecord::Base
  acts_as_taggable
end
```

```
david    = Person.find(1)
welcome  = Message.find(1)
summer   = Tag.find_or_create_by_name "Summer"

summer.on(david, welcome)

david.tags    # => [ summer ]
welcome.tags  # => [ summer ]

summer.tagged # => [ david, welcome ]
```

```
class CreateTalks < ActiveRecord::Migration
  def self.up
    create_table :talks do |t|
      t.string :title
      t.integer :author_id
      t.datetime :starts_at
      t.datetime :ends_at

      t.timestamps
    end
  end

  def self.down
    drop_table :talks
  end
end
```

```
msch@iMartin.stuwo5: ~/Desktop/jsug — bash
bash
Last login: Mon Mar  2 17:24:03 on ttys003
cd msch@iMartin:~ $ cd Desktop/jsug/
msch@iMartin:~/Desktop/jsug $ rake db:migrate
(in /Users/msch/Desktop/jsug)
== CreateTalks: migrating =====
-- create_table(:talks)
   -> 0.0023s
== CreateTalks: migrated (0.0025s) =====

== CreateAuthors: migrating =====
-- create_table(:authors)
   -> 0.0022s
== CreateAuthors: migrated (0.0024s) =====

msch@iMartin:~/Desktop/jsug $ █
```

```
set :application, "backpack"
set :repository,  "svn+ssh://somewhere/nice"

set :gateway, "gateway.37signals.com"

role :app, "app1.37signals.com", :primary => true
role :app, "app2.37signals.com"
role :app, "app3.37signals.com"
role :web, "web1.37signals.com", :primary => true
role :web, "web2.37signals.com"
role :db,  "db1.37signals.com",  :primary => true
```

TIM



02.03.09 16:31:24

Willkommen !**Status****You are not logged in!****Menu**

- ☐ [Home](#)
- ☐ [Schwarzes Brett](#)
- ☐ [Zeitplan](#)
- ☐ [Downloads](#)
- ☐ [F.A.Q.](#)

- ☐ [Login](#)
- ☐ [Registrierung
EPROG](#)

FastLogin

Matrikelnummer:

Passwort:

Links[EPROG](#)[PROLOG](#)[zur Instituts Seite](#)**Wiki:EprogHome****"EINFÜHRUNG IN DAS PROGRAMMIEREN", 183.210, VL (VORMALS 183.046, AU)**

Willkommen bei der Lehrveranstaltung **"Einführung in das Programmieren"** (EPROG) . EPROG wird als Vorlesung mit Laborübung (VL 6.0ECTS/4.0SWH) mit der LvaNr: 183.210 vom Institut für Rechnergestützte Automation angeboten.

EPROG ist eine Pflichtlehrveranstaltung für StudienanfängerInnen der Bakkalaureatsstudien Informatik und des Bakkalaureatsstudiums Wirtschaftsinformatik.

LEHRINHALT UND LEHRZIELE

Inhalt der Lehrveranstaltung ist eine Einführung in grundlegende Konzepte einer modernen objekt-orientierten Programmiersprache und deren Umsetzung anhand der Programmiersprache Java. Konkrete Themen sind:

- Grundstrukturen, Variablen, Entwerfen von Algorithmen
- Klassen und Objekte
- Methoden und Parameter
- Interfaces und Vererbung
- Containerklassen und Arrays
- Rekursives Programmieren

Die TeilnehmerInnen lernen, grundlegende Methoden des strukturierten und objekt-orientierten Programmierens durch praktische Programmierbeispiele umzusetzen. Sie sollen in die Lage versetzt werden, weitere Details der Sprache Java selbst zu erarbeiten und auch andere verwandte Programmiersprachen zu erlernen.

AKTUELLE INFORMATION

Alle wichtigen aktuellen Informationen (Vorbesprechung, Termine, etc) finden Sie am **Schwarzen Brett**

VORBESPRECHUNG ZU "EINFÜHRUNG IN DAS PROGRAMMIEREN" 183.210

Die Vorbesprechung für EPROG im Sommersemester 2008 findet am **Mittwoch, 4. März 2009** von 11h c.t. bis ca. 12h im **E17** statt.

TIM Beta

Einführung in das Programmieren

Martin Schürer (0620611)

Lehrveranstaltungen


Logout

Übersicht

Aufgaben

Anmeldungen

Benotung

 Abgabe wird bewertet

Aufgaben

Hier findest du eine Übersicht deiner aktuellen Aufgaben.

Runde 1 — Hello World!

Fällig seit 1 day (28 Feb 17:19)

Grüßen Sie Ihr erstes Programm in Java nicht nur die ganze Welt
sonder auch Ihre Familie

1 von 10 Versuchen hochgeladen.

Ihre Abgabe befindet sich momentan in der Warteschlange.



 [Komplette Angabe](#)
[Durchgeführte Abgaben](#)

Runde 2 — Mixins

Fällig in 1 day (03 Mar 17:19)

Grüßen Sie Ihr erstes Programm in Java nicht nur die ganze Welt sonder
auch Ihre Familie

0 von 10 Versuchen hochgeladen.

 [Komplette Angabe](#)
 [Lösung abgeben](#)

✓ Sie wurden erfolgreich zu dem Termin March 10, 2009 17:19 - 18:19 angemeldet

Anmeldungen

Für diese Termine bist du angemeldet.

Datum	Raum	Termin
March 10, 2009 17:19 - 18:19	HGEG01	1. Prüfung Details Ab-/Ummelden

Verfügbare Termine

Diese Termine sind momentan für dich verfügbar.

Datum	Termin	Platz
March 21, 2009 17:19 - March 31, 2009 17:19	1. Abgabegespräch	25 Plätze frei Details Anmelden
March 10, 2009 17:19 - March 20, 2009 17:19	1. Prüfung	Angemeldet Details Ab-/Ummelden

TIM Beta

Einführung in das Programmieren

Martin Schürer (0620611)

Lehrveranstaltungen

Logout

Übersicht

Aufgaben

Anmeldungen

Benotung

Ab-/Ummelden

Abmelden

Datum

10 Mar 16:19 - 17:19

Ummelden

10 Mar 17:19 - 18:19

10 Mar 18:19 - 19:19

Ummelden

10 Mar 19:19 - 20:19

Ummelden

10 Mar 20:19 - 21:19

Ummelden

10 Mar 21:19 - 22:19

Ummelden

10 Mar 22:19 - 23:19

Ummelden

10 Mar 23:19 - 00:19

Ummelden

11 Mar 00:19 - 01:19

Ummelden

11 Mar 01:19 - 02:19

Ummelden

TIM Beta

Einführung in das Programmieren

Tim Tutor (tutor1)

Lehrveranstaltungen

Logout

Übersicht

Interviews

Anstehende Interviews

Zeit	Student	Termin	
21 Mar 16:19	Max Muster (230044)	1. Abgabegespärch	Durchführen
21 Mar 16:19	Irfan Adilovic (131740)	1. Abgabegespärch	Durchführen
21 Mar 16:19	Martin Schürer (25649)	1. Abgabegespärch	Durchführen
21 Mar 17:19	Thomas Schranz (197789)	1. Abgabegespärch	Durchführen

TIM Beta

Einführung in das Programmieren

Tim Tutor (tutor1)

Lehrveranstaltungen

Logout

Übersicht

Interviews

Anstehende Interviews

Zeit	Student	Termin	
21 Mar 16:19	Max Muster (230044)	1. Abgabegespärch	Fortsetzen Abbrechen
21 Mar 16:19	Irfan Adilovic (131740)	1. Abgabegespärch	Durchführen
21 Mar 16:19	Martin Schürer (25649)	1. Abgabegespärch	Durchführen
21 Mar 17:19	Thomas Schranz (197789)	1. Abgabegespärch	Durchführen

Abgabegespräch

Runde 1 — Hello World!

Fällig: 04 Mar 17:18

Grüßen Sie Ihr erstes Programm in Java nicht nur die ganze Welt
sonder auch Ihre Familie

 [Angabe](#)[Durchgeführte Abgaben](#)

1 von 10 Versuchen hochgeladen.

Der [1. Versuch](#) erreichte 20 von 20 Punkten.

2. Versuch

[Download](#)

Eingabe	Erwartet	Ausgabe
4 4 +	8	8
4r4 +	FALSCHE EINGABE	FALSCHE EINGABE
4 4 +	8	8
4 4 -	0	0
4 4 *	16	16
4 4 /	1	1

Martin Schürer (25649)

Runde 2 — Interfaces World!

Fällig: 03 Mar 17:18

Grüßen Sie Ihr erstes Programm in Java nicht nur die ganze Welt sonder auch Ihre
Familie

 [Angabe](#)

0 von 10 Versuchen hochgeladen.

```
specification.rb (~/projects/ti...cise_chains/a/01helloworld) - VIM2
1 use :io_pair
2
3 submission :prepare => 'javac *.java', :exec => 'java Calculator'
4 reference :check_for => 'Calculator.class'
5
6 operations = ['+', '-', '*', '/']
7
8 test do
9   iopair '4 4 +'
10  iopair '4r4 +'
11
12  operations.each do |op|
13    iopair "4 4 #{op}"
14  end
15 end
~
~
~
~
~
~
~
specification.rb 7,0-1 All
```


Lessons learned

Avoid Second-System Syndrome
If in doubt leave it out!

KISS Keep it Simple, Stupid

YAGNI You Ain't Gonna Need It

Release Early, Release Often

Use **git** ;)

When to choose Rails and when not to

Starting from a **clean slate**

WebAPP not WebSITE

CRUD Create Read Update Delete

RAD Rapid Application Development

Rails is **Opinionated Software**

Ruby

GET

POST

PUT

DELETE

find

create

update

destroy

SELECT

INSERT

UPDATE

DELETE

POST /people/create

GET /people/show/1

POST /people/update/1

POST /people/destroy/1

POST	/people
GET	/people/1
PUT	/people/1
DELETE	/people/1

```
ActionController::Routing::Routes.draw do |map|  
  map.resources :person  
end
```

```
class PeopleController < ActionController::Base
  # POST /people
  def create() end

  # GET /people/1
  def show() end

  # PUT /people/1
  def update() end

  # DELETE /people/1
  def destroy() end
end
```

```
class PeopleController < ActionController::Base
  # POST /people
  def create() end

  # GET /people/1
  def show() end

  # PUT /people/1
  def update() end

  # DELETE /people/1
  def destroy() end
end
```


form_for :person, david do |f|
 f.text_field :name
end


```
class PeopleController < ActionController::Base
  # POST /people
  def create() end

  # GET /people/1
  def show() end

  # PUT /people/1
  def update() end

  # DELETE /people/1
  def destroy() end
end
```



A diagram consisting of two orange rectangular boxes connected by a horizontal line. The left box contains the code `# GET /people/1` and `def show() end`. The right box contains the code `link_to "David",` and `person_url(david)`. The line connects the `show()` method to the `link_to` call.

```
class PeopleController < ActionController::Base
  # POST /people
  def create() end

  # GET /people/1
  def show() end

  # PUT /people/1
  def update() end

  # DELETE /people/1
  def destroy() end
end
```

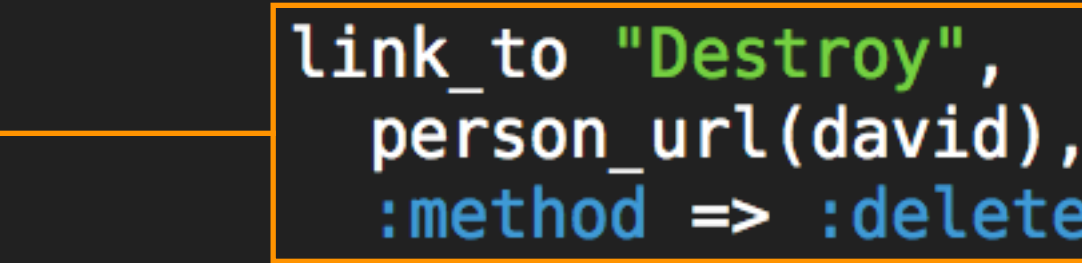
```
form_for :person, david,
  :method => :put do |f|
  f.text_field :name
end
```

```
class PeopleController < ActionController::Base
  # POST /people
  def create() end

  # GET /people/1
  def show() end

  # PUT /people/1
  def update() end

  # DELETE /people/1
  def destroy() end
end
```



A horizontal line connects the `def destroy() end` line in the code to a box containing its implementation. The box contains the following code:

```
link_to "Destroy",
  person_url(david),
  :method => :delete
```

```
class PeopleController < ActionController::Base
  # GET /people
  def index() end

  # GET /people/new
  def new() end

  # POST /people
  def create() end

  # GET /people/1
  def show() end

  # GET /people/1;edit
  def edit() end

  # PUT /people/1
  def update() end

  # DELETE /people/1
  def destroy() end
end
```

```
class Account < ActiveRecord::Base
  def upgrade_to(plan)
    if eligible_for_upgrade?(plan)
      update_attribute :plan_id, plan
      true
    else
      false
    end
  end
end
```

```
class Plan < ActiveRecord::Base
end
```

```
class AccountController < ActionController::Base
  def upgrade_plan
    if @account.upgrade_to(Plan[params[:plan_name]])
      redirect_to :action => "index"
    else
      flash[:notice] = "Account not eligible for upgrade"
      render
    end
  end
end
```

```
class Account < ActiveRecord::Base
  has_many :subscriptions
end

class Subscription < ActiveRecord::Base
  belongs_to :account
  belongs_to :plan

  protected
  def validate_on_create
    unless eligible_for_upgrade?
      add_to_base("Account not eligible for upgrade")
    end
  end
end

class Plan < ActiveRecord::Base
end

class SubscriptionsController < ActionController::Base
  def create
    @account.subscriptions.create(:plan => Plan[params[:plan_name]]) ?
      redirect_to(:action => "index") :
      render
  end
end
```