Ruby on Rails

Фрэймворк для разработки
Web-приложений

Сайты

- Twitter
- Github
- Yellowpages.com
- ……

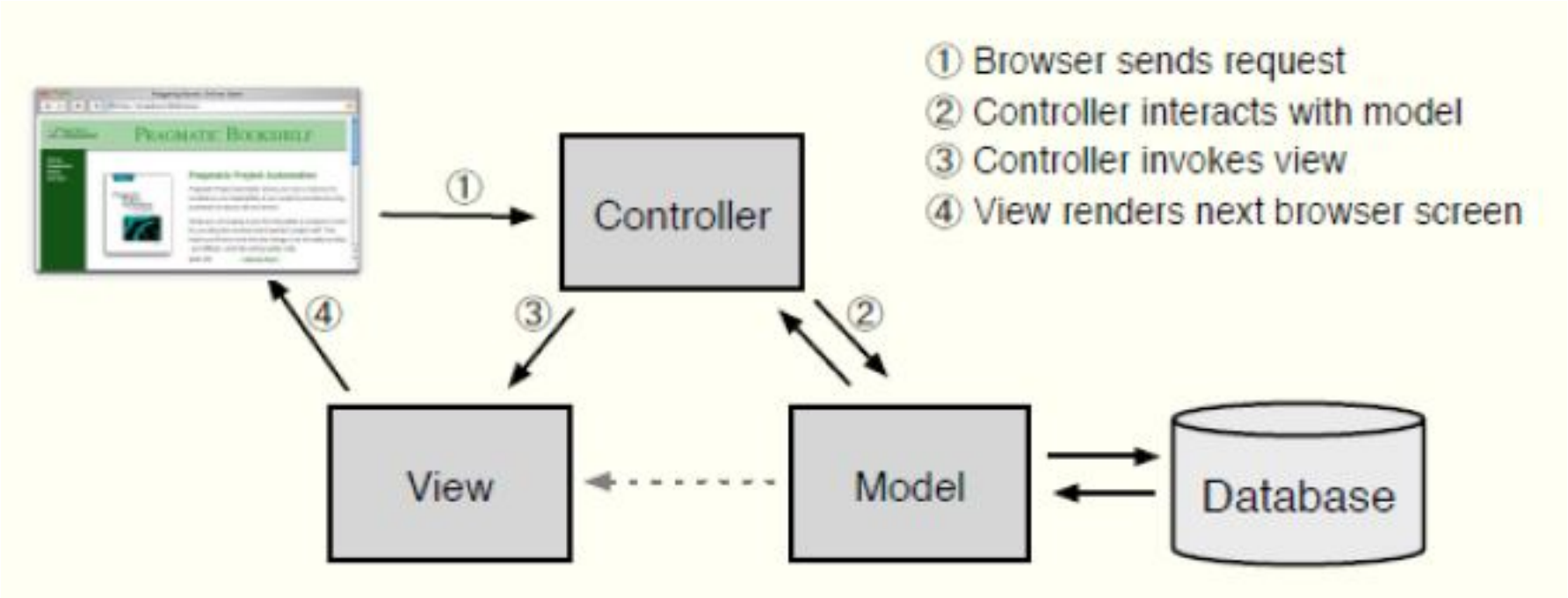
Ruby on Rails

- Красота и мощь языка Ruby
- Реализация паттерна MVC
- Встроенный сервер *WEBrick*
- Поддержка баз данных: *MySQL, PostgreSQL, SQLite, SQL Server, DB2, Oracle*
- Принцип соглашений вместо конфигурирования
- Роутинг URL
- Встроенная система тестирования
- База готовых решений (плагинов)

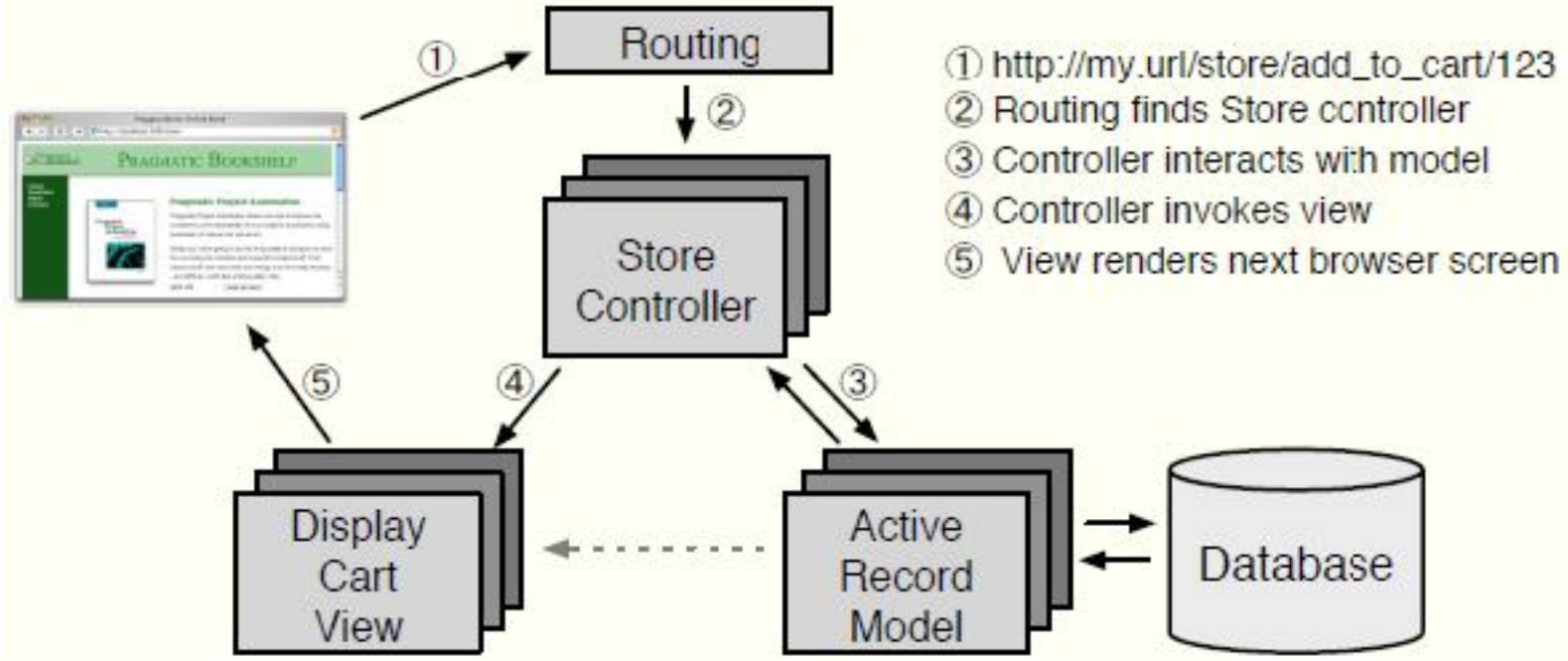
Принципы

- Архитектура MVC
- Соглашение по конфигурации (соглашение по именованию и расположению файлов)
- Отсутствие повторов в коде (DRY).
- Быстрая разработка (agile development)
  - Автоматизированное тестирование кода приложения.
  - Рефакторинг.

MVC


① Browser sends request
② Controller interacts with model
③ Controller invokes view
④ View renders next browser screen
Controller
View
Model
Database

MVC


Routing
Store Controller
Display Cart View
Active Record Model
Database
① http://my.url/store/add_to_cart/123
② Routing finds Store controller
③ Controller interacts with model
④ Controller invokes view
⑤ View renders next browser screen

MVC

- **M**odel (**Модель**) является «сутью» приложения и отвечает за непосредственные алгоритмы, расчёты и тому подобное внутреннее устройство приложения. Также предоставляет линк к хранилищу данных.
- **V**iew (Представление, **Вид**) предназначен для вывода данных, предоставленных Моделью. Это единственная часть MVC, которая непосредственно контактирует с пользователем.
- **C**ontroller (Поведение, **Контроллер**) получает данные от пользователя и передаёт их в Модель. Кроме того, он получает сообщения от Модели и передаёт их в Вид.

Структура Rails-приложения

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
components/
config/
db/
 - migrate/
doc/
lib/
public/
script/
test/
 - fixtures/
 - functional/
 - integration/
 - mocks/
 - unit/
tmp/
vendor/
```

```
>rails conference
    …
    …
    create  doc/README_FOR_APP
    create  log/server.log
    create  log/production.log
    create  log/development.log
    create  log/test.log
```

Структура Rails-приложения

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
```

- **app**: динамическая часть вашего приложения
  - controllers: интерфейс и деловая логика
  - helpers: код для поддержки Видов
  - models: объекты Модели, соотносящиеся с БД и используемые в приложении
  - views: шаблоны для отсылки удаленному пользователю
  - layouts: файлы макетов страниц

```
config/
 - environment.rb
 - routes.rb
 - database.yml
```

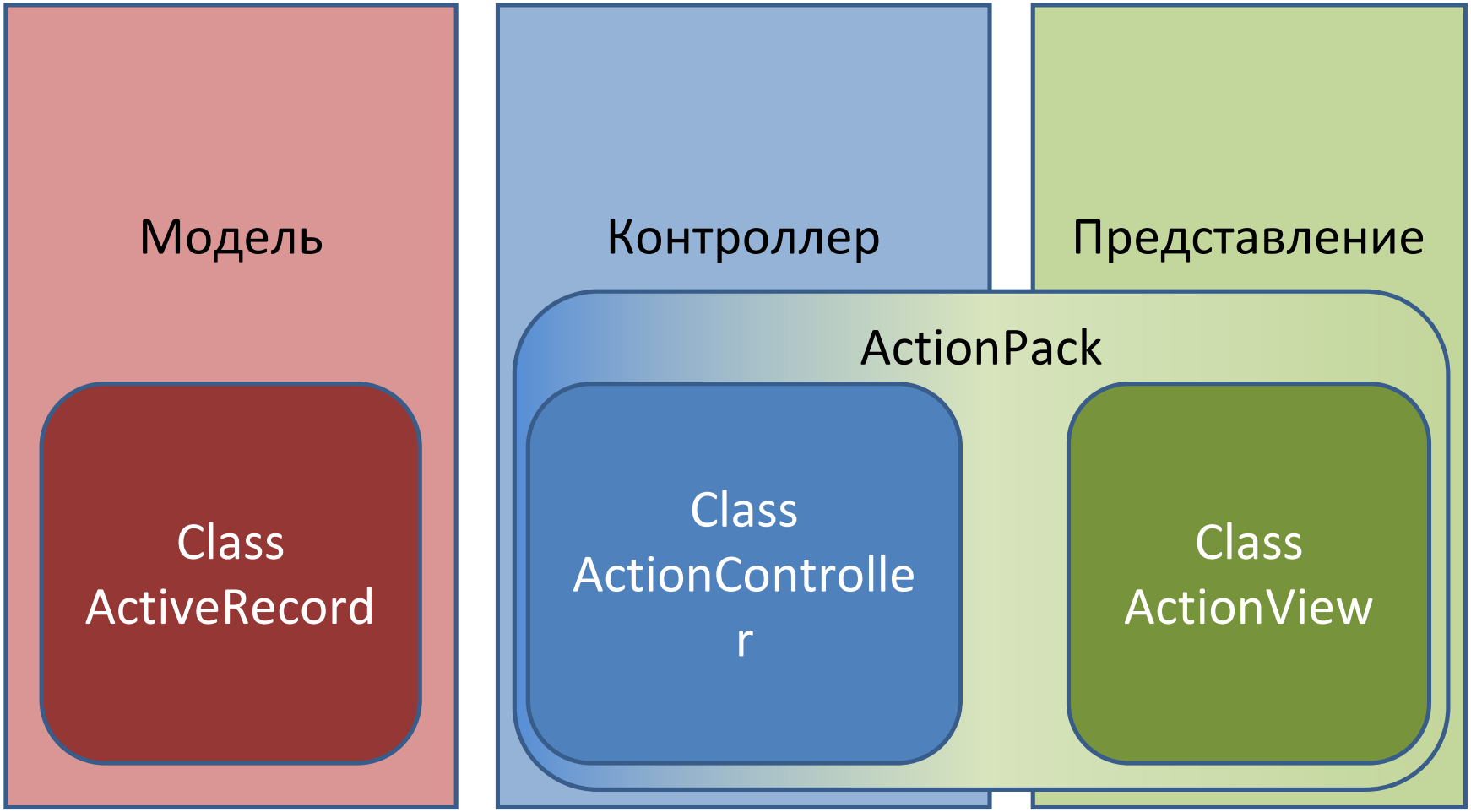
database.yml

```
development:
  adapter: mysql
  database: conference_development
  username: root
  password:
  host: localhost

test:
  adapter: mysql
  database: conference_test
  username: root
  password:
  host: localhost

production:
  adapter: mysql
  database: conference_production
  username: root
  password: change_me
  host: localhost
```

Модули Ruby on Rails / MVC



ActionPack

ActionPack отвечает за уровни контроллера и представления

conferees_controller.rb

```
class ConfereeController < ApplicationController
    def index
        @people = ‘SPbAU SE'
        @message = 'Это сообщение пришло из Контроллера'
    end
end
```

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
```

index.rhtml

```
<html>
    <head><title>Hi <%= @people %>!</title></head>
    <body>
    <h1>Hello!</h1>
    <p>Это приветствие прибыло</p>
    <p><%= @message %></p>
    </body>
</html>
```

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
    - conferees/
```

ActionPack

index.rhtml

```
<html>
    <head><title>Hi <%= @people %>!</title></head>
    <body>
        <h1>Hello!</h1>
        <p>Это приветствие прибыло</p>

        <% 5.times do%>
        <p><%= @message %></p>
        <% end %>
    </body>
</html>
```

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
    - conferees/
```

ActiveRecord

Таблицы:
- conferences
- conferees
- papers

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
```

conference.rb

```
class Conference < ActiveRecord::Base
end
```

conferee.rb

```
class Conferee < ActiveRecord::Base
end
```

paper.rb

```
class Paper < ActiveRecord::Base
end
```

```
>> rupy = Conference.new(:name => "RuPy Omsk 2007", :date => "2007-02-10")
=> #<Conference:0x322f610 @attributes={"name"=>"RuPy Omsk 2007",
"date"=>"2007-02-10"}, @new_record=true>
>> rupy.save
=> true
>> Paper.find(:all)
=> #Array of all papers from db
```

ActiveRecord - ассоциации

has_one
has_many
belongs_to
has_many_and_belongs_to

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
```

conference.rb

```
class Conference < ActiveRecord::Base
  has_many :conferees
end
```

conferee.rb

```
class Conferee < ActiveRecord::Base
  belongs_to :conference
  has_one :paper
end
```

paper.rb

```
class Paper < ActiveRecord::Base
  belongs_to :conferee
end
```

```
ivanov = Conferee.new(:name => "Ivanov Vasily", :city => "Omsk")
=> #<Conferee:0x31feed8 @attributes={"city"=>"Omsk", "name"=>"Ivanov
Vasily", "email"=>nil, "conference_id"=>nil}, @new_record=true>
rupy.conferees << ivanov
ivanov.conference.name
=> "RuPy Omsk 2007"
ivanov.paper = Paper.new(:title => "Ruby and Python comparison", :duration =>
25)
ivanov.paper.title
=> "Ruby and Python comparison"
```

Проверка данных на корректность

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
```

```
class Conferee < ActiveRecord::Base
  validates_presence_of  :name, :email
  validates_format_of :email,
         :with =>/^\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,3}$/
end
```

```
ivanov = Conferee.new(:name => "Ivanov Vasily", :city => "Omsk")
=> #<Conferee:0x31feed8 @attributes={"city"=>"Omsk", "name"=>"Ivanov
Vasily", "email"=>nil, "conference_id"=>nil}, @new_record=true>
ivanov.save
=> false
ivanov.errors.on :email
=> "can't be blank"
ivanov.email = "ivanov#mail.ru"
ivanov.save
=> false
ivanov.errors.on :email
=> "is invalid"
```

Проверка данных на корректность

Существующие методы

validates_acceptance_of
validates_associated
validates_confirmation_of
validates_exclusion_of
validates_format_of
validates_inclusion_of
validates_length_of
validates_numericality_of
validates_presence_of
validates_size_of
validates_uniqueness_of

Для нетривиальных случаев

validate
validate_on_create
validate_on_update

```
class Conferee < ActiveRecord::Base
  def validate_on_create
    unless knows_ruby_and_python?(self)
      errors.add("Conferee", "needs to learn both ruby and python")
    end
  end
end
```

Обратные вызовы

```
class CreditCard < ActiveRecord::Base
  def before_validation_on_create
    self.number = number.gsub(/[^0-9]/, "") if attribute_present?("number")
  end
end
```

```
class Subscription < ActiveRecord::Base
  before_create :record_signup

  private
    def record_signup
      self.signed_up_on = Date.today
    end
end
```

before_validation
before_validation_on_create
after_validation
after_validation_on_create
before_save
before_create
after_create
after_save

Транзакции

```
transaction do
  kostia.send(100)
  pasha.receive(100)
end

Account.transaction(kostia, pasha)
  kostia.send(100)
  pasha.receive(100)
end
```

Используются для того, чтобы предотвратить изменения в БД, в случае, если одно из действий не удается выполнить

Чтобы не допустить изменения объектов, следует использовать транзакции уровня объектов

Migrations

create_table
drop_table
add_column
remove_column
change_column

db/
 - **migrate**/

001_initial_setup.rb

```
class InitialSetup < ActiveRecord::Migration

  def self.up
    create_table(:conferences) do |t|
      t.column :name,          :string
      t.column :date,          :date
    end
    create_table(:conferees) do |t|
      t.column :name,          :string
      t.column :email,         :string
      t.column :city,          :string
      t.column :conference_id, :integer
    end
    create_table(:papers) do |t|
      t.column :title,         :string
      t.column :start_at,      :time
      t.column :duration,      :integer
      t.column :conferee_id,   :integer
    end
  end

  def self.down
    [:conferences,:conferees,:papers].each {|t| drop_table t}
  end
end
```

Migrations

```
>rake db:migrate
(in R:/conference)
== InitialSetup: migrating ==========
-- create_table(:conferences)
   -> 0.1110s
-- create_table(:conferees)
   -> 0.1200s
-- create_table(:papers)
   -> 0.1200s
== InitialSetup: migrated (0.3610s) =
```

Мигрэйты позволяют поддерживать данные в базе в актуальном состоянии при изменении структуры таблиц, предоставляют возможность отката до любого предыдущего состояния.

ActionPack

ActionPack отвечает за уровни контроллера и представления

conferees_controller.rb

```
class ConfereesController < ApplicationController
  def index
    @conferees = Conferee.find(:all)
  end
end
```

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
```

index.rhtml

```
<% unless @conferees.empty? %>
<ul>
  <% @conferees.each |c|%>
    <li><%= c.name %><%= "(#{c.city})" if c.city %></li>
  <% end%>
</ul>
<% end %>
```

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
    - conferees/
```

- Ivanov Vasily (Omsk)
- Makarov Ivan

Представления – Views

```
<h2><%= @conferee.name %></h2>
<h3><%= @conferee.city if @conferee.city %></h3>
<div><%= render :partial => "paper_details" %></div>
```

Partials

_paper_details.rhtml

```
<h4><%= @conferee.paper.title %></h4>
<p><%= @conferee.paper.description %></p>
```

Layouts

```
app/
 - controllers/
 - helpers/
 - models/
 - views/
    - layouts/
```

```
<html>
  <head>
    <title><%= @page_title %></title>
  </head>
  <body>
    <div>Шапка макета</div>
    <div><%= @content_for_layout %></div>
    <div>Подвал макета</div>
  </body>
</html>
```

Scaffolding

```
class ConfereesController < ApplicationController
  scaffold :conferee
end
```

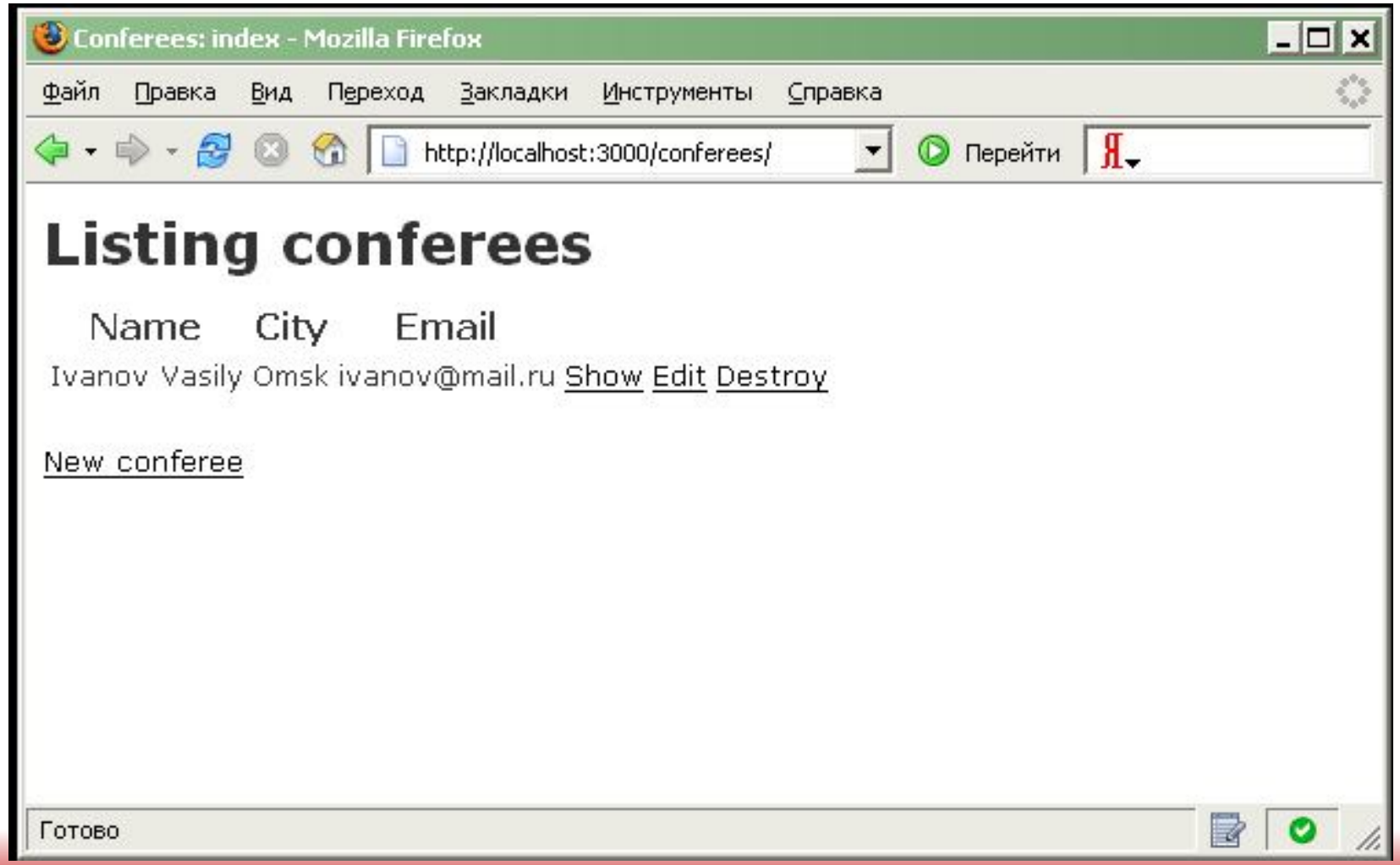
Динамический скаффолдинг

```
R:\conference>ruby script/generate scaffold conferee
    ...
    create  app/views/conferees/_form.rhtml
    create  app/views/conferees/list.rhtml
    create  app/views/conferees/show.rhtml
    create  app/views/conferees/new.rhtml
    create  app/views/conferees/edit.rhtml
    create  app/controllers/conferees_controller.rb
    create  test/functional/conferees_controller_test.rb
    create  app/helpers/conferees_helper.rb
    create  app/views/layouts/conferees.rhtml
    create  public/stylesheets/scaffold.css
```

Статический скаффолдинг
- генерирует набор файлов, которые вы можете переписать или дописать под свои нужды

Scaffolding

```
>ruby script/server
=> Booting WEBrick...
=> Rails application started on http://0.0.0.0:3000
=> Ctrl-C to shutdown server; call with --help for options
```



Helpers - помощники

```
<%= link_to "Профиль Иванова", :controller => 'conferees',
                               :action => 'show', :id => 1 %>

<% form_tag(:action => 'update', :id => @conferee) do %>
  <%= text_field_tag "name", @conferee.name %>
  <%= text_area_tag "description", @conferee.description %>
  <%= submit_tag 'Edit' %>
<% end %>
```

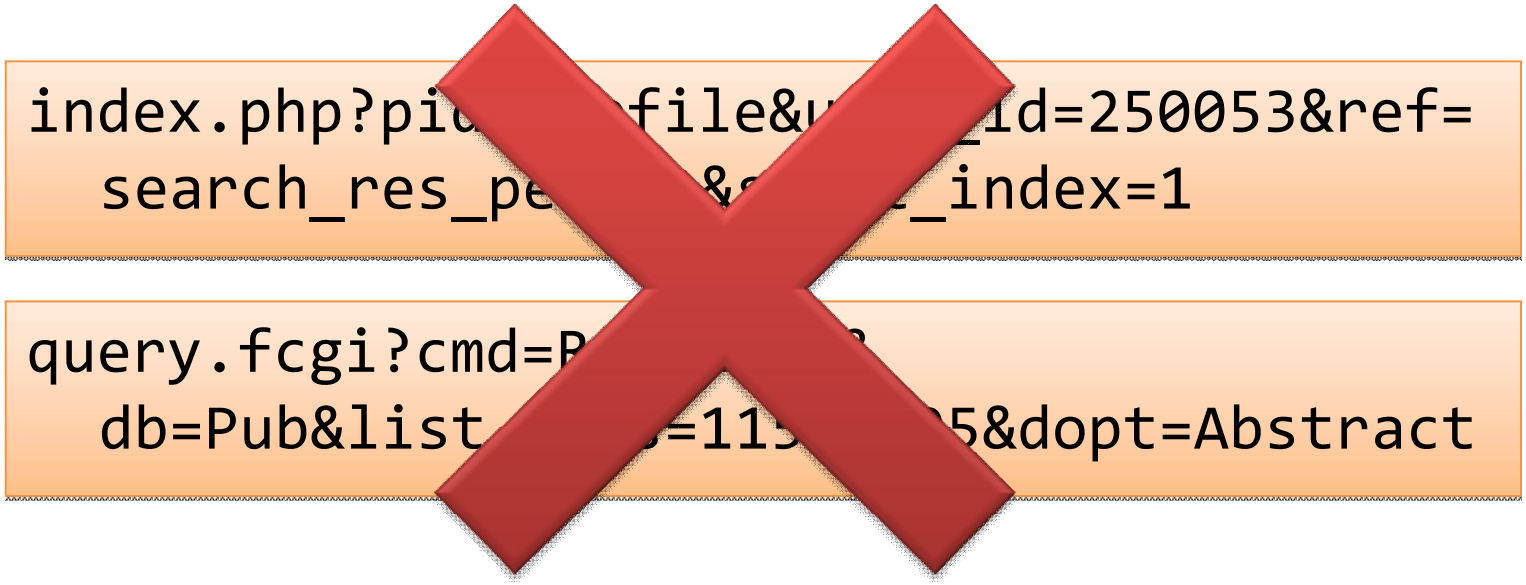
```
app/
 - controllers/
 - helpers/
 - models/
 - views/
```

```
<a href="/conferees/show/1">Ivanov</a>

<form action="/conferees/update" method="post">
  <input id="name" name="name" type="text" value="Ivanov"/>
  <textarea id="description" name="description"></textarea>
  <input name="commit" type="submit" value="Edit" />
</form>
```

```
index.php?pid=profile&user_id=250053&ref=
  search_res_people&start_index=1
```

```
query.fcgi?cmd=Retrieve&
  db=Pub&list_uids=11579295&dopt=Abstract
```



/books/ruby/1

/blog/2007/2/10/rupy

/conferees/ivanov/edit

/cars/mazda/5/buy

/news/2006/11/02

/news/rss

Роутинг URL

```
map.connect ':controller/:action/:id’
```

```
config/
  - routes.rb
```

```
map.connect 'conferees/:city', :controller => 'conferees', :action => 'by_city'
```

```
/conferees/omsk
```

```
map.connect 'rupy', :controller => 'conferences', :action => 'show', :id => 1
```

```
/rupy
```

```
map.connect 'admin/:controller/:action/:id', :controller => /conferences|conferees/
```

```
/admin/conferees/delete/13
```

ActionMailer

```
def signup_notification(recipient)
  recipients recipient.email_address_with_name
  subject "Информация о новом аккаунте"
  from "system@example.com"

  attachment :content_type => "image/jpeg", :body =>
   File.read("an-image.jpg")

  attachment "application/pdf" do |a|
    a.body = generate_your_pdf_here()
  end
end
```

```
deliver_signup_notification(ivanov.email)
```

Автоматизированое тестирование в Rails

```
test/
 - unit/
 - fixtures/
 - functional/
 - mocks/
 - integration/
```

Unit-тесты – тестирование моделей

Fixtures – исходные данные для тестов

Функциональные тесты – проверка работы контроллеров

fixture conferees.rb

```
ivanov:
  id: 1
  name: Ivanov Vasily
  email: ivanov@mail.ru
  conference_id: 1

petrov:
  id: 2
  name: Petrov Egor
  email: petrov@rambler.ru
  conference_id: 1
```

Mocks – заглушки для реальных объектов, недоступных при тестировании

Интеграционные тесты – реализация User Stories, проверка работы контроллеров в комплексе

Автоматизированое тестирование в Rails

```
assert_equal 2, 1+1              assert_valid
assert_not_nil something         assert_select
assert_kind_of String            ...
assert_response
```

```
def goes_to_admin_only_page
   get '/admin/index'
   assert_response :redirect
   assert_redirected_to '/account/login'
   post '/account/login', {:login => "test", :password => "test"}
   assert_response :redirect
   assert_redirected_to '/admin/index'
   follow_redirect!
   assert_template 'admin/index'
   assert_select 'h3', 'New profiles'
   assert_select 'table#new_profiles' do
     assert_select 'table.profile_summary', 2
   end
 end
```

Плагины

Google Map Plugin

Acts as Authenticated

Acts as Rateable

Acts as Audited

Acts as Versioned

Acts as State Machine

Globalize

Active Form

Acts as Ferret

Juggernaut

Markaby

file_column

```
class Conferee < ActiveRecord::Base
  file_column :photo
end
```

```
<%=file_column_field "conferee", "photo"%>
```

```
<%=image_tag url_for_file_column("entry", "image")%>
```

Интеграция с RMagick

```
class Conferee < ActiveRecord::Base
  file_column :photo, :magick => {
     :versions => { "thumbnail" => "48x48", "portrait" => "600x800>" }
    }
end
```

acts_as_taggable

```
ruby ./script/plugin install acts_as_taggable
```

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

```
paper = Paper.new(:title=>"Ruby & Python comparison")
paper.tag_with("ruby python")
paper.save
paper.tag_list
# => "ruby python"

Paper.find_tagged_with("ruby")
# => массив докладов с тагом ruby
```

acts_as_list
acts_as_tree
acts_as_nested_set
Observers
Фильтры
Кэширование
serialize
Работа с XML
polymorphic assoc.

AJAX, RJS, prototype
respond_to
Aggregations
REST
Отладка, breakpoints
ActionWebService
ActiveSupport
и еще много всего :)