使用 Ruby On Rails 15分钟开发一个博客系统

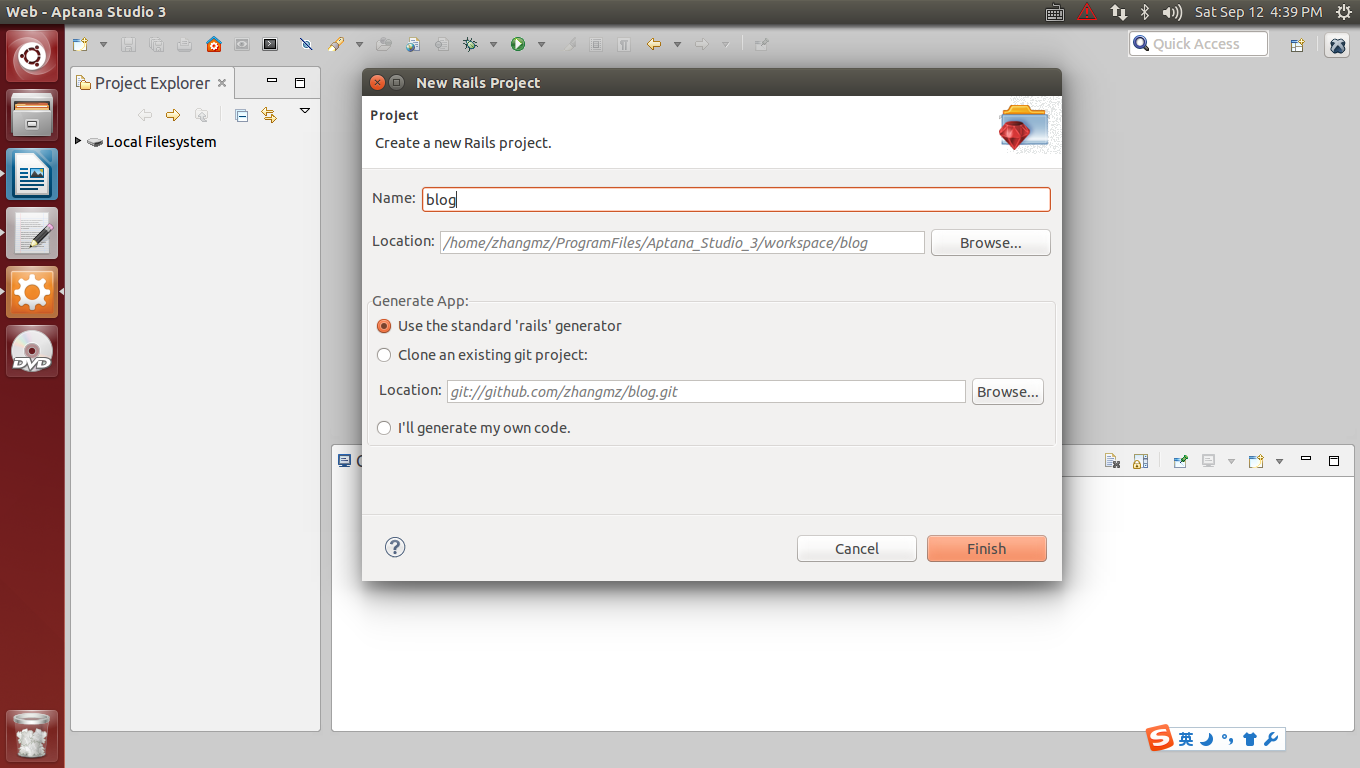
作者：张孟志

日期：2015-09-13

联系：[zhangmengzhi2005@126.com](mailto:zhangmengzhi2005@126.com)

说明：Ruby on Rails 的入门示例。Ubuntu 14.04/Ruby 1.9.3/Rails 4.2

1、使用IDE（Aptana Studio 3）创建一个博客项目；



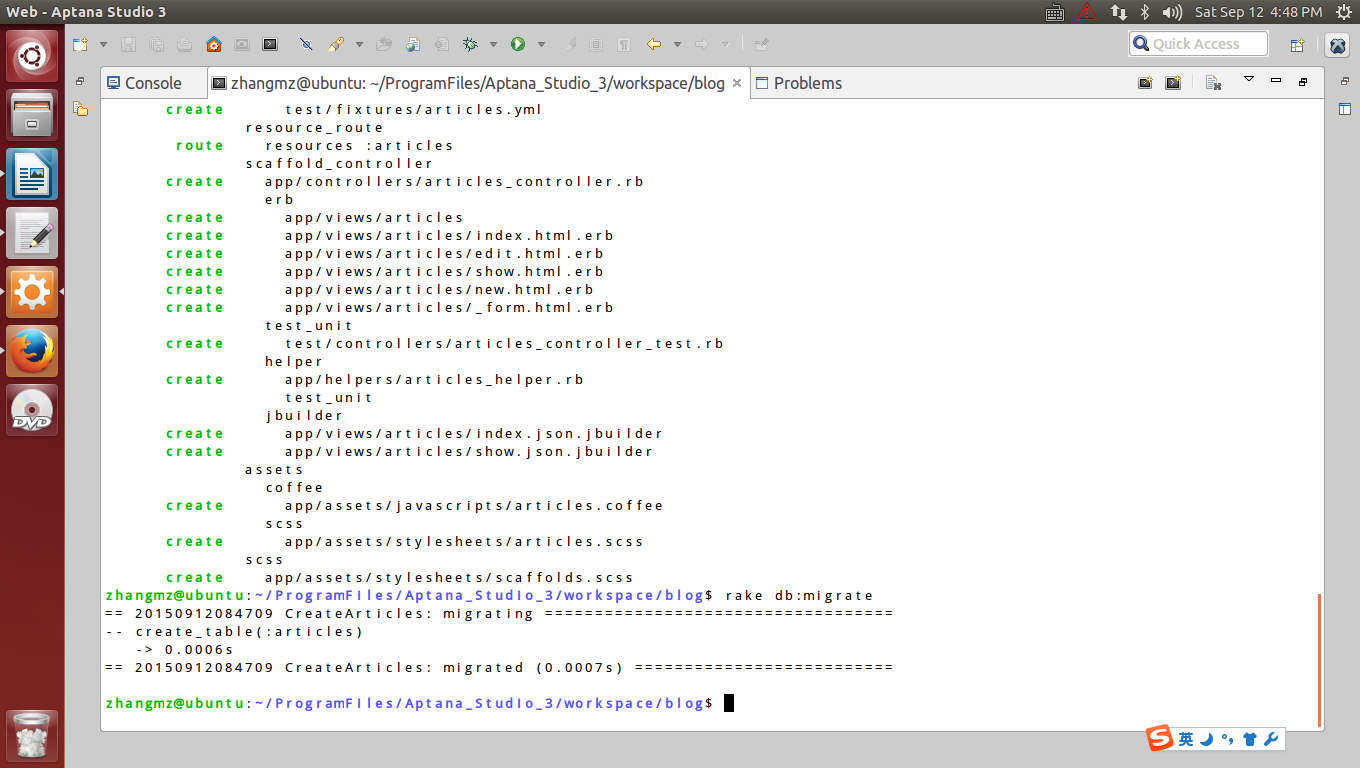
2、在终端窗口创建一个博客的支架；

博客文章保存到 Article ，包含标题、内容两部分。

保证你的路径在博客项目下，例如：/home/zhangmz/ProgramFiles/Aptana\_Studio\_3/workspace/blog。

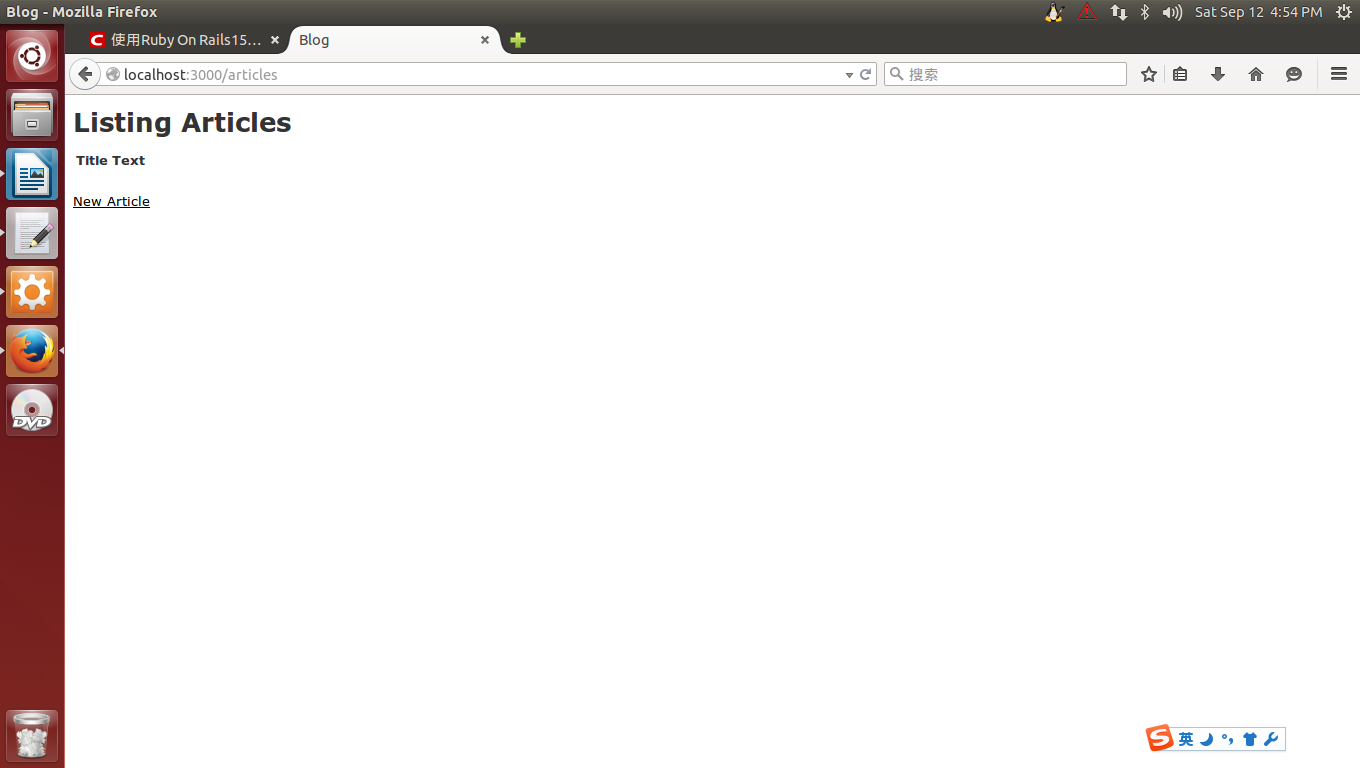
rails generate scaffold Article title:string text:text

rake db:migrate

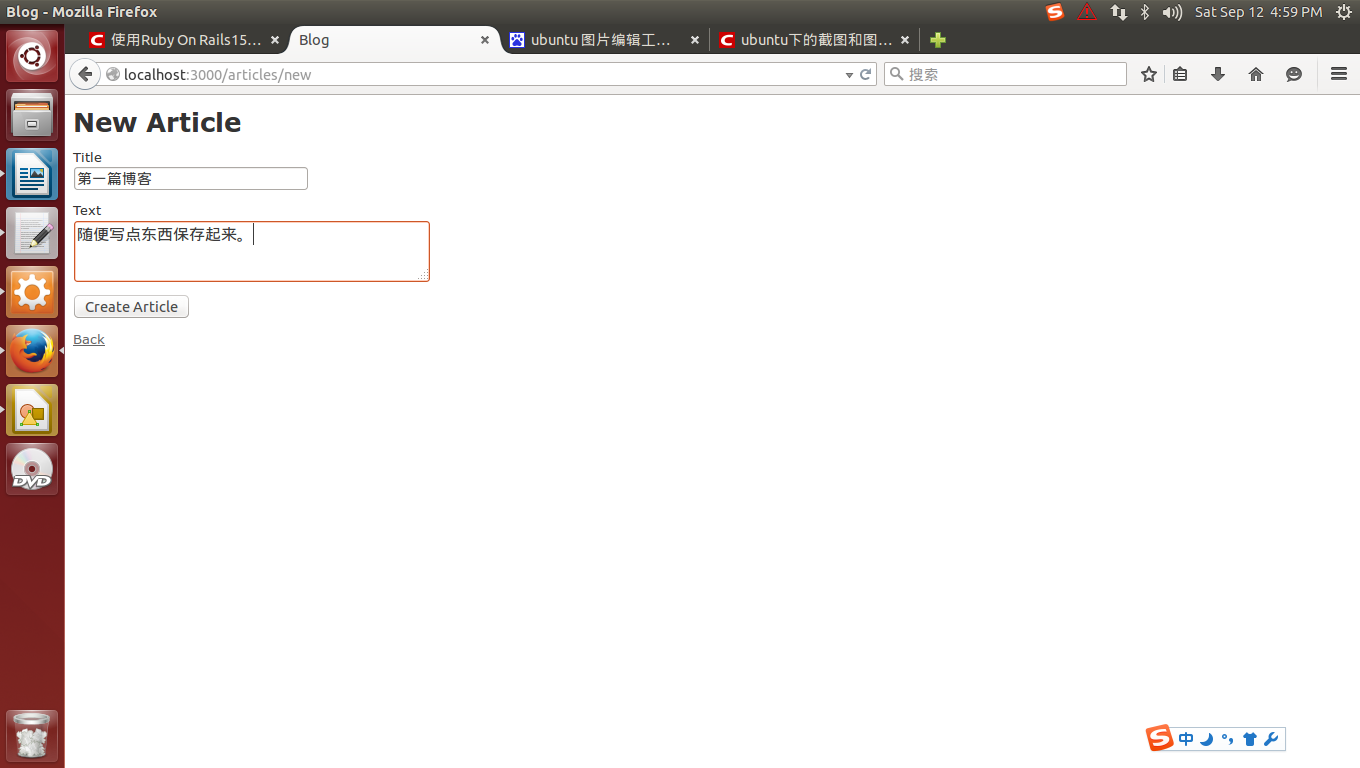


3、启动应用；

rails s



4、测试 博客项目的Article支架；



5、简单调整一下页面；

修改blog/app/views/articles/index.html.erb

<p id**=**"notice">

**<%=** notice **%>**

</p>

<h1>Listing Articles</h1>

**<%** **@articles.**each **do |**article**| %>**

<h2>**<%=** link\_to article**.**title**,** article**%>**</h2>

<p>

<i>-**<%=** time\_ago\_in\_words article**.**created\_at **%>** ago </i>

</p>

<p>

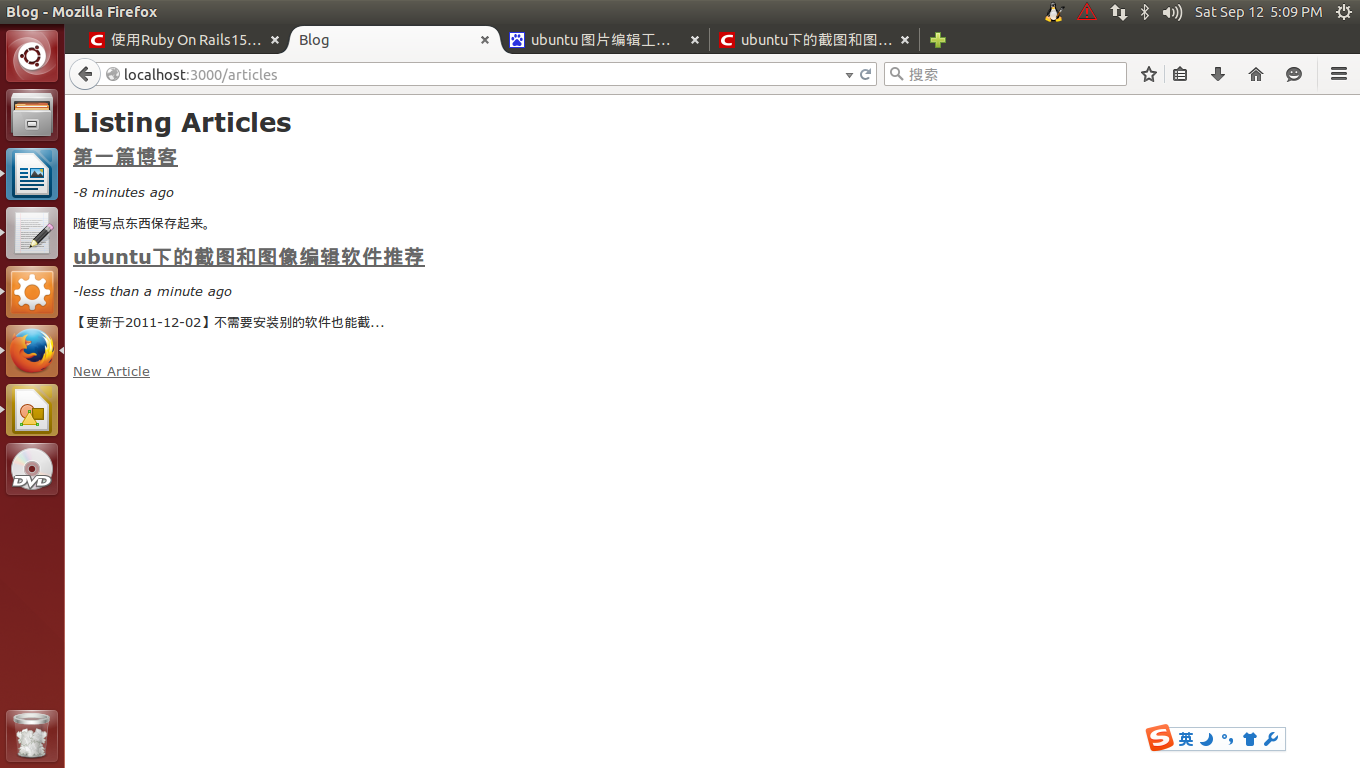
**<%=** truncate article**.**text **%>**

</p>

**<%** **end %>**

<br>

**<%=** link\_to 'New Article'**,** new\_article\_path **%>**



修改blog/app/views/articles//show.html.erb

<p id**=**"notice">**<%=** notice **%>**</p>

<p>

<strong>Title:</strong>

**<%=** **@article.**title **%>**

</p>

<p>

<strong>Text:</strong>

**<%=** **@article.**text **%>**

</p>

**<%=** link\_to 'Back'**,** articles\_path **%>** |

**<%=** link\_to 'Edit'**,** edit\_article\_path**(@article) %>** |

**<%=** link\_to "Delete"**,@article,:method => :delete, data: { confirm: "Are you sure?" }** **%>**



6、添加评论；

评论必然是针对某一特定的博客，因此，评论应该包括一个博客文章的ID作为外键关联，另外评论设计为一个文本。评论是跟随博客文章展示的，不需要View，我们单独创建model/controller。

rails generate model Comment article\_id:integer text:text

rake db:migrate

rails generate controller Comments create destroy

7、在博客文章下面添加评论；

修改blog/app/views/articles//show.html.erb，用于评论显示/发布/删除。

<p id**=**"notice">

**<%=** notice **%>**

</p>

<p>

<strong>Title:</strong>

**<%=** **@article.**title **%>**

</p>

<p>

<strong>Text:</strong>

**<%=** **@article.**text **%>**

</p>

<h2>Comments</h2>

**<%** **@article.**comments**.**each **do |**comment**| %>**

<p>

**<%=** comment**.**text **%>**

</p>

<p>

**<%=** time\_ago\_in\_words comment**.**created\_at **%>** ago &nbsp;&nbsp;&nbsp;&nbsp;

**<%=** link\_to "Delete Comment"**, [@article,** comment**], :method => :delete,** data: **{** confirm: "Are you sure?" **} %>**

</p>

**<%** **end %>**

**<%=** form\_for **[@article,@article.**comments**.**build**] do |**f**| %>**

<p>

**<%=** f**.**text\_area **:text, :size =>** '20x5' **%>**

</p>

<p>

**<%=** f**.**submit "Post Comment" **%>**

</p>

**<%** **end %>**

**<%=** link\_to 'Back'**,** articles\_path **%>** |

**<%=** link\_to 'Edit'**,** edit\_article\_path**(@article) %>** |

**<%=** link\_to "Delete"**,@article,:method => :delete,** data: **{** confirm: "Are you sure?" **} %>**

修改blog/config/routes.rb，让博客文章加载评论。

resources **:articles do**

resources **:comments**

**end**

修改blog/app/models/article.rb，令博客文章关联评论。

**class** Article < ActiveRecord**::**Base

has\_many **:comments,** dependent: **:destroy**

**end**

修改blog/app/models/comment.rb，令博客文章关联评论，同时检验评论不能为空。

**class** Comment < ActiveRecord**::**Base

belongs\_to **:article**

validates **:text,** presence: true

**end**

修改blog/app/controllers/comments\_controller.rb，完成评论的创建与删除。

**class** CommentsController < ApplicationController

**def** create

**@article** = Article**.**find**(**params**[:article\_id])**

**@comment** = **@article.**comments**.new(**comment\_params**)**

**@comment.**save

redirect\_to **@article**

**end**

**def** destroy

**@article** = Article**.**find**(**params**[:article\_id])**

**@comment** = **@article.**comments**.**find**(**params**[:id])**

**@comment.**destroy

redirect\_to **@article**

**end**

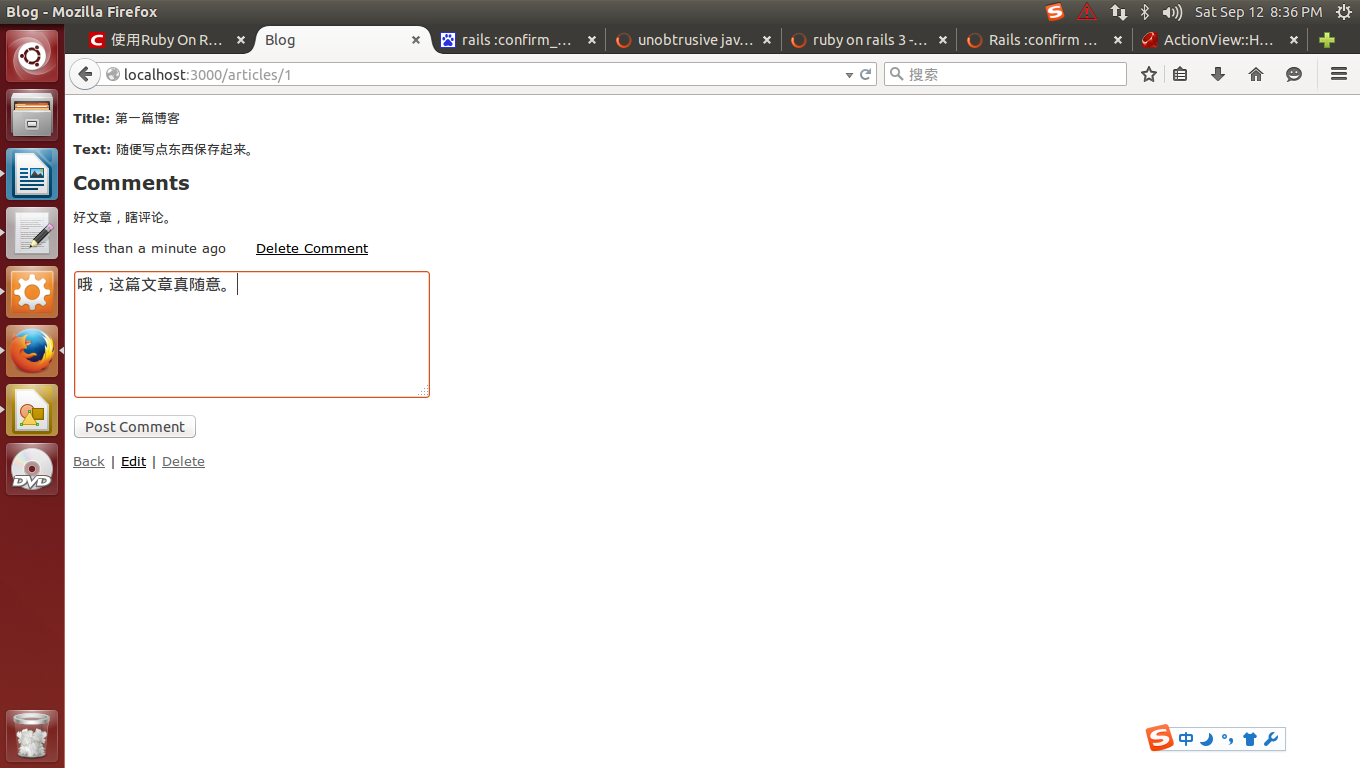
private

**def** comment\_params

params**.**require**(:comment).**permit**(:post\_id , :text)**

**end**

**end**



其它：

1、查看sqlite数据库。

sqlite3 -line db/development.sqlite3

.quit

2、使用Bootstrap装饰。

安装Bootstrap：sudo gem install bootstrap-sass

Gemfile加入Bootstrap：gem 'bootstrap-sass'

blog/assets/javascripts/application.js添加：//= require bootstrap

blog/config/application.rb在class Application < Rails::Application一行下添加一行：config.assets.paths << "#{Rails}/vendor/assets/fonts"

创建blog/app/assets/stylesheets/custom.css.scss：

@import "bootstrap"**;**

body **{**

padding-left**:** 60px**;**

**}**