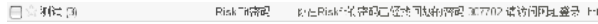


rails send mail set from name

效果:



以前 用 google 的 smtp 服务 成功发送了 email ,

<http://www.blog.java.net/f11429/archive/2009/05/04/268866.html>

由于 google 的 smtp 每天只容许 发送 500 封email, 所以必须自己搭建邮件服务器了。。。。邮件服务器用的是 kerio mail server , 搭好之后 enviroment.rb 中 配置

```
ActionMailer::Base.perform_deliveries = true
ActionMailer::Base.raise_delivery_errors = true
ActionMailer::Base.default_charset = "utf-8"
ActionMailer::Base.default_content_type = "text/html"
ActionMailer::Base.smtp_settings = {
  :address => "59.314.13.266",
  :port => 25,
  :domain => "www.abc.com",
  :user_name => "webmaster",
  :password => "riskfit654321",
  :authentication => :login
}
```

Model里的send 方法

```
def send_password(recipient, subject, name,password)
  @subject = subject
  @recipients = recipient
  @from = 'webmaster@abc.com'
  @sent_on = Time.now
  @body['name'] = name
  @body['password'] = password
  @headers = {}
end
```

可是 发送 email 后 , 邮箱 显示的发件人 始终是 webmaster , 那么我想让发件人的名字显示成 测试 , 该如何设置呢, 那么把@from 设置成这样 :

```
@from = '"测试"<webmaster@abc.com>' # 主意是外面是单引号 , 里面 是 双引号 , 而且用户名webmaster 必须和 config里的user_name 一样
```

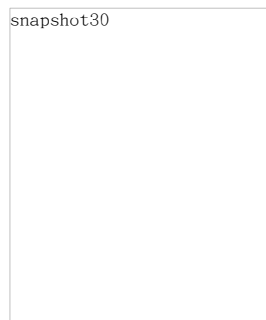
ref:

<http://www.javaeye.com/topic/126875>

Rails 自引用实现最近来访

环境 : ruby 1.8.6 + rails 2.1.0 + ubuntu 8.10

效果: 就像 xiaonei 这样



当然最简单的办法 你可以 直接sql操作就ok 了。。。我这里介绍 利用 rails的 activerecord自引用 解决, 一个系统有一个users表, 但是每一个user可能有friends 或者 followers , 而这个 friends 和 followers 也在 users这个表里, 这样就需要 用到 rails的自引用了。

实现最近来访 只是实现了一层的 相当于 取出 twitter的 followers

表结构:

1, 用户表 client\_infos (username,password,.....)

2, 访问关系表 vistorships

snapshot31

model关系

```
class ClientInfo < ActiveRecord::Base
  has_many :vistorships
  has_many :vitors, :through => :vistorships
end

class Vistorship < ActiveRecord::Base
  belongs_to :client_info
  belongs_to :vistor, :class_name => "ClientInfo"
end
```

添加最近来访 数据:

```
@client_info = ClientInfo.find(params[:id])
@recentvistor = @client_info.vistorships.build( :vistor => current_client_info)
@recentvistor.save
```

取出全部来访 数据

```
@client_info.vistorships
```

Model 取出最近来访的12个vistor

```
def get_vitors(client_info) # client_info 是显示那个人的对象
  find(:all,
    :conditions => ["client_info_id = ? and vistor_id < ?", client_info.id, client_info.id],
    :select => ["distinct(vistor_id)"],
    :limit => 12,
    :order => 'created_at DESC'
  )
end
```

controller中取出数据

```
@latestvitors = Vistorship.get_vitors(@client_info)
```

View中输出

```
<ul>
  <%= for lvistor in @latestvitors -%>
<li>
  <%= image_link lvistor.vistor, :image => :thumbnail , :class => 'image' %>
  <%= client_info_link lvistor.vistor, :class => 'name' %>
</li>
  <%= end -%>
</ul>
```

ref:

<http://railscasts.com/episodes/163-self-referential-association>

**Rails in place edit with jQuery**

环境 : ruby 1.8.6 + rails 2.1.0 + jquery 1.3.2

效果:

<http://www.appelsiini.net/projects/jeditable/default.html>

<http://15daysofjquery.com/examples/jqueryEditInPlace/divEdit.php>

像上面的这种 in place edit 的效果在网络上很常见。。下面介绍一种在rails中的解决方案。。当然后台不一定是 ror 平台了

资源下载:

<http://github.com/tuupola/jquery.jeditable> 下载后, 只需要里面的 jquery.jeditable.js 步骤:

1,加载 jquery.js 和 jquery.jeditable.js

2,View

```
<script>
$(document).ready(function() {
  $("edit_descrip").jeditable(this.href, {
    indicator : "<img src='/images/indicator.gif'>",
    type : 'textarea',
    submitdata: { _method: "get" },
    select : true,
    submit : '修改',
    cancel : '取消',
    cssclass : "editable",
    tooltip : '点击修改',
    id : 'flag'
  });
});
</script>
<p class="edit_descrip" id="edit_descrip" onmouseover="this.style.backgroundColor='yellow'" onmouseout="this.style.backgroundColor='white'">%= h @client_info.description %> </p>
```

默认向后台发送的是id 和 value 两个参数, 值分别是p对应的id 和 p 对应的内容, 当然你也可以改变参数名字

```
id : 'flag'
value : 'newvalue'
```

就对应传递 flag 和 newvalue 两个参数 了

3,Controller:

```
if request.xhr?
  if params[:flag]
    case params[:flag]
      when 'edit_descrip'
        ClientInfo.update_field(current_client_info.id,params[:value],'descrip') # 更新数据库
        render :text => params[:value] #返回数据
    end
  end
end
```

4,改变一些默认的参数, 可以到 jquery.jeditable.js 中修改, 例如 textarea 的默认大小 等等。。。。

5,或者你可以指定target为某个函数, 这样就可以自己扩展, 例如有额外的返回数据, 返回后需要更新某个 DOM元素等等

```
<script>
$(document).ready(function() {
  $("edit_state").jeditable(submitEdit, {
    indicator : "<img src='/images/indicator.gif'>",
    type : 'textarea',
    submitdata: { _method: "get" },
    select : true,
    submit : '修改',
    cancel : '取消',
    cssclass : "editable",
    tooltip : '点击修改',
    id : 'flag'
  });
});

function submitEdit(value, settings)
{
  var edits = new Object();
  edits[settings.name] = [value];
  edits[settings.id] = ['edit_state']
  $.ajax({
    url: this.href,
    type: "GET",
    data : edits,
    dataType : "json",
    complete : function ()
    {
      $("#state_time").html('0分钟前');
    }
  });
  return value;
}
</script>
```

See:

<http://www.appelsiini.net/projects/jeditable>

<http://stackoverflow.com/questions/966539/how-to-get-out-of-jquery-jeditable-mess>

simple jQuery date-picker plugin 中文版

效果:



汉化了一个 jquery calendar plugin , 效果还符合 中国人的使用习惯。。。

使用步骤:

1, 加载需要的js 和 样式

```
<link href="calendar.css" rel="stylesheet" type="text/css">
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="cal.js"></script>
```

2, 使用

```
jQuery(document).ready(function () {
    S('input.one').simpleDatepicker(); // 最基本的
    S('input.two').simpleDatepicker({ startdate: 2008, enddate: 2012 }); // 定义年的范围, 一般用这个
    S('input.three').simpleDatepicker({ chosendate: '9/9/2010', startdate: '6/10/2008', enddate: '7/20/2011' }); //定义当前选择的日期
    S('input.four').simpleDatepicker({ x: 45, y: 3 }); //定义日历基于input的显示位置
});
</script>
<p><input class="one" name="date" value="" type="text">
<input class="two" name="date" value="" type="text">
<input class="three" name="date" value="" type="text">
<input class="four" name="date" value="" type="text">
</p>
```

DEMO:

<http://wxianfeng.com/demos/calendar>

source:

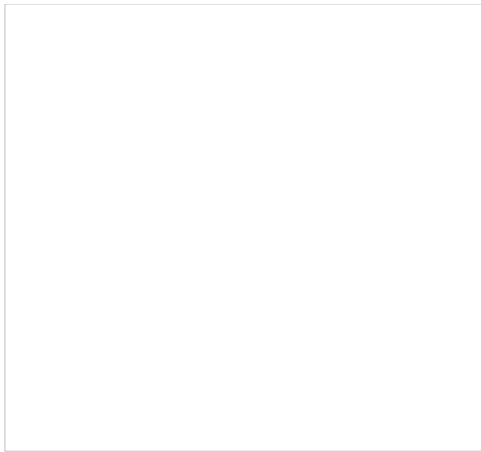
<http://www.uushare.com/filedownload?user=f11429&id=2183466>

ref:

<http://teddevito.com/demos/calendar.php>

<http://www.aspxhome.com/javascript/skills/20099/1008917.htm>

SNS : renren.com 的盈利模式



renren.com 是目前属于国内一个大型的社区网站之一了，背后有着众多的大学生群体，包括已经毕业了，走上社会的大学生。

renren.com 原先 是 xiaonei.com 个人还是喜欢 xiaonei.com 校内网。。名字好听，而且调查xiaonei的用户后，发现大多数人仍然喜欢输入xiaonei.com打开校内网，而不是renren.com。从xiaonei刚出来的时候，我就用了，一直用到现在，可以说是看着xiaonei一步一步成长起来的，和xiaonei同时出来的还有 zhanzuo.com（占座网），估计05~07界的学生都知道当时学校 贴的海报 宣传的 xiaonei 和 zhanzuo。

记得在08年的时候日本软银 收购 xiaonei 40%的股权，给予 4.5 亿美元的融资，当时在学生中还引起不小的轰动，好多人都注销了xiaonei，以抵抗 小日本 SNS盈利模式一直被人质疑，今天闲来无事，想看下校内的盈利模式，于是就 瞅 了一下，发现xiaonei赚钱的地方还真不少。

#### 赚钱一：广告

目前xiaonei的主要收入来源估计还是广告，打开xiaonei 可以发现 有好多 大的企业在上边投放 广告，例如我现在打开xiaonei首页，有 仁和药业 集团， 凡客诚品 等

#### 赚钱二：增值服务

什么是增值服务，增值服务就是有超过了自己网站的范畴，额外的服务的意思在里面，例如xiaonei推出的 web game： 猫游记，通过这个 web game，可以在里面植入广告，虚拟物品购买，交易等

#### 赚钱三： 开放平台

xiaonei 后来学习了facebook，推出了 开放平台，xiaonei 为了鼓励人们开发xiaonei的app，给予这样的奖励措施，app没邀请一个星级用户，可以得到2元的奖励，还有和 google adsense 合作，按照流量进行广告分成。

这样有了开放平台后，就会 开发出很多的 第三方app 了，不清如此，最重要的是和成熟的第三方应用合作，例如 很火的开心农场，这样通过开心农场收入的钱，由两家公司分成，另外xiaonei 也可以在 开心农场中植入自己的广告，来赚钱

#### 赚钱四：虚拟礼品

虚拟物品在别的社区网站可能没有生存的价值，可还在xiaonei网还是生机勃勃的，因为xiaonei是一个真实资料的社区，一般都是自己同校的，或者以前的同学，甚至是情侣，所以喜欢送礼物的人 大有人在，每送一个礼品，消耗一个 校内豆，一个xiaonei豆 是 一元

#### 赚钱五： 星级用户

星级用户即付费用户，xiaonei里叫紫豆会员，就像qq会员一样，有一些特殊的权限，例如有很多好的皮肤给你选择，发言没有限制，不是星级用户会出现验证码等等

#### 赚钱六： google adsnse广告

流量有了，相信google adsnse 也会带来不小的收入的。。。。

更多的盈利模式，还需要 发现的眼睛。。。

rails comet juggernaut

环境 : ruby 1.8.7 + rails 2.1.0 + ubuntu 8.10 desktop

comet 是 server push 技术,说白了就是 服务器端直接 推送数据到客户端,据说 是 未来能取代ajax的一门技术, comet要求客户端 和 server 要建立一个长连接,无论是http 方式 还是 socket 方式, ajax 是单用户的异步请求, comet是多用户异步请求, 这个技术 已经应用的很广了, 特别对于一些 交互性 和 实时性 要求比较高的 系统,例如 股票 实时刷新, web聊天 (web qq,gtalk,meebo.com等等), 实时提醒(xiaonei.com) 等功能,

那么 在rails中 有这样的comet现成东西吗? 答案是有, juggernaut 插件, juggernaut 原理是什么呢, 官网是这样 说的:

- 1, 客户端A 和 comet server 建立一个 socket 连接 (注意这里的server 是 juggernaut的 server)
- 2, 客户端B 向 rails server 发送一个 异步 ajax 请求 (例如这里的rails server 开发环境下为 webrick)
- 3, rails server 发送数据 给comet server
- 4, comet server 广播数据到 客户端 (客户端用户可以指定)

plugin 依赖的 库 :

- \* Rails 2.0.2 or edge
- \* json gem (gem install json)
- \* EventMachine gem (gem install eventmachine)
- \* juggernaut gem (gem install juggernaut)

demo: 实时聊天室

基于 prototype || jQuery

- 1, 新建 rails project
- 2, 安装插件

```
script/plugin install http://juggernaut.rubyforge.org/svn/trunk/juggernaut
```

安装完后 会在 config 下看到 juggernaut 的 host 文件

- 3, 配置 host 文件

```
:hosts:
- :port: 5001 # 默认就是5001
  :host: 192.168.1.3 # 你电脑的ip
  :public_host: 192.168.1.3
  :public_port: 5001
```

- 4, cd 到 config 目录下, 生成 juggernaut 的 配置文件

```
juggernaut -g juggernaut.yml
```

会在 config下 看到 juggernaut.yml 文件

- 4, 配置 juggernaut.yml

```
:allowed_ips:
- 127.0.0.1
- 192.168.1.3 # 添加你电脑的ip
```

- 5,开启 juggernaut

```
juggernaut -c juggernaut.yml
```

显示成 Starting Juggernaut server 0.5.8 on port: 5001... 这样就 已经开启了, 不要以为还在 start 当中

- 6,controller

```
class ChatController < ApplicationController
```

```

def index
end

def chat_prototype
end

def chat_jquery
end

# prototype
def send_data_p
  render :juggernaut do |page|
    page.insert_html :top, 'chat_data', "<li>#{h params[:chat_input]}</li>"
  end
  render :nothing => true
end

# jQuery
def send_data_j
  render :juggernaut do |page|
    page["#chat_data"].prepend "<li>#{h params[:chat_input]}</li>"
    # page["#titl"].Text = "wang"
    page["#updatetitle"].click()
  end
  render :nothing => true
end
end

```

## 7, View

chat\_prototype.html.erb (基于 prototype )

```

<html>
<head>
  <title>测试</title>
  <%= javascript_include_tag :defaults, :juggernaut %>
  <%= juggernaut(:debug => false) %>
</head>
<body>
  <%= form_remote_tag(
    :url => { :action => :send_data_p },
    :complete => "S('chat_input').value = ''" ) %>
  <%= text_field_tag( 'chat_input', '', { :size => 20, :id => 'chat_input' } ) %>
  <%= submit_tag "Add" %>
</form>
<ul id="chat_data" style="list-style:none">
</ul>
</body>
</html>

```

上面的 就是 和 comet server 建立连接的, 还有很多参数可以添加, 详细到 juggernaut helper 文件中查看, 例如我上面加了 :debug => false 就是 不输出 调试

chat\_jquery.html.erb (基于jQuery, 这个我给它加了动态改变title 的做法, 原先想在action中改的, 可是没成功, 后来就在action中click页面button, 然后button 触发js 修改title 成功)

```

<html>
<head>
  <title id="titl" runat="server">测试</title>
  <%= javascript_include_tag 'jquery', 'json', 'juggernaut/juggernaut', 'juggernaut/jquerynaut', 'juggernaut/swfobject' %>
  <%= juggernaut(:debug => false) %>
</head>
<body>
  <form action="" method="get">
    <div style="margin:0;padding:0">
      <input id="chat_input" name="chat_input" size="20" type="text" value="" />
      <input name="commit" type="submit" value="Add" />
    </div>
  </form>

  <script>
    S(document).ready(function() {

      S('form').submit(function() {
        S.get('/chat/send_data_j', { chat_input: S('#chat_input').val() })
        return false;
      })
    })

    function setttitle() {
      var b = "新提醒";
      var c = "::-:";
      var t = new Date();
      s = t.getSeconds();
      if(s%2 == 0){
        document.title = b;
      }else{
        document.title = c;
      }
      setTimeout("setttitle()", 1000);
    }
  </script>

```

```

<div style="display:none">
  <input type="button" id="updatetitle" onclick='settitle();' value='Change Title' />
</div>

<ul id="chat_data" style="list-style:none"></ul>

</body>
</html>

```

注意jquery 加载的资源文件，和 prototype 的 不一样，所有资源在下载 的 插件media中 可以找到

8,开启webrick

9,打开网址开始聊天

[http://192.168.1.3:3000/chat/chat\\_jquery](http://192.168.1.3:3000/chat/chat_jquery)

多个用户 打开上面的网址就可以实时聊天了.....

ref:

<http://juggernaut.rubyforge.org/>

<http://ajaxian.com/archives/juggernaut-comet-for-rails>

<http://macrochen.javaeye.com/blog/28020>

09-10 赛季 火箭队赛程 短信提醒



效果预览:(html地址)

<http://www.google.com/calendar/embed?src=i0ihq25ni844u94gbkq0k44btcj8arf3%40import.calendar.google.com&ctz=Asia/Shanghai>

我们知道利用google calendar 的免费 reminder(email,SMS,pop-up) 功能，可以给你的日常事物 做很多 有用的事情，例如：

- 1，上课课表，设置 短信 提前提醒
- 2，家人的生日 提前提醒
- 3，设定重要事情 提醒
- 4，中国 重要节气的提醒



5, 。 。 。 。 。

这里介绍给喜欢NBA的朋友，设置 火箭比赛 提前 短信提醒。。。, 有一点, 是英文的, 不过 这 提醒已经足够提醒你哪天 哪一时间 有 火箭 比赛了。。

使用要求:

- 1,有一个 google 帐号
- 3,一个手机号码 (联通 移动 皆可)

使用办法:

- 1,登录 google 打开 calendar
- 2, 打开yahoo的calendar , yahoo已经制作好了,  
<http://sports.yahoo.com/nba/teams/hou/schedule>

点击 Add to calendar -> 选择 google calendar

- 3,这样 就自动加入到 你的 google calendar 中了

- 4, 把你的 google calendar 时区设置成 北京时间, 这样 显示 就是 中国 的比赛时间了

- 5, 最后 一步 设置 短信 提醒, ok。。等着 收短信吧。

以上步骤比较 笼统, 更详细的 google it, 很多

## Jquery IE 没有权限



环境 : jquery 1.3.2

最近在用 jquery + will\_paginate 实现 ajax 分页 , 按照 我以前文章里的办法<http://www.blog.java.net/f11429/archive/2009/08/25/292522.html>发现在IE下 出现 没有权限 的错误, 也就 看不到 ajax 效果了。。

在网上搜了一下, 出现这个 错误, 说是 跨域问题, 可是 url 里又没有涉及到 跨域的情况, 不管它 了, 后来 把url 的绝对路径 , 改成了 相对路径, ie 下 就没出现 这个 错误了, 怪~!

will\_paginate 生成的 href 代码是这样的, 可以看出 是 相对路径

```
<a rel="next" href="/user/172/stateposts?page=2">2</a>
```

原先代码 :

```
jQuery(function() {  
    jQuery(".my_paginate a").live("click", function() {  
        jQuery(".my_paginate").html("正在加载...");  
        jQuery.get(this.href, {flag : "my" }, null, 'script');  
        return false;  
    });  
});
```

```
});
```

重点是 `this.href` 得到的是绝对路径 例如<http://abc.com/user/show> 这样的路径

修改后代码

```
jQuery(function() {  
    jQuery(".my_paginate a").live("click", function() {  
        jQuery(".my_paginate").html("正在加载...");  
        jQuery.get(jQuery(this).attr("href"), {flag : "my" }, null, 'script');  
        return false;  
    });  
});
```

发现 把 `url` 变成了 `jQuery(this).attr("href")` 这个 得到的 相对路径 ,例如 `/user/show`

如果你在 `ie` 下 遇到了 , 没有权限错误, 我想 肯定 也是 请求的路径 (相对, 绝对) 问题了。。

ref:

<http://www.2ky.cn/h/12/4716.html>