

Rails #8 (补充)

架构/环境/安全观念等

架构设计

- development
- test
- production

Development

- ▶ 开发用
- ▶ 变数不 cache
- ▶ 故障时可直接看到程式码 debug

Development

NoMethodError in Admin::ProductsController#new

undefined method `admin?' for #<User:0x007f98d95a5ef0>

Extracted source (around line #9):

```
7   def admin_required
8
9     if !current_user.admin?
10       redirect_to "/"
11     end
12   end
```

Production

- 正式环境
- 变数会被 cache
- 出错只会给你看 500 画面

We're sorry, but something went wrong.

We've been notified about this issue and we'll take a look at it shortly.

环境变量

- config/environments/**development.rb**
- config/environments/**production.rb**

环境变量

config/environments/development.rb

```
Rails.application.configure do
  ... (略)
  + config.after_initialize do
  +   Pay2go.integration_mode = :development
  + end
end
```

config/environments/production.rb

```
Rails.application.configure do
  ... (略)
  + config.after_initialize do
  +   Pay2go.integration_mode = :production
  + end
end
```


在 Controller 内判断环境

```
def mail_to_customer
  if Rails.env == "development"
    do_xxx
  else
    do_yyy
  end
end
```

安全观念

密码不要放在 .rb 档案里面

其实这是很危险的

```
Pay2go.setup do |pay2go|  
  if Rails.env.development?  
    pay2go.merchant_id = "1234567" # 放测试站的 key  
    pay2go.hash_key    = "xxxxxxxx"  
    pay2go.hash_iv     = "xxxxxxxx"  
    pay2go.service_url = "https://capi.pay2go.com/MPG/mpg_gateway"  
  else  
    pay2go.merchant_id = "1234567" # 放正式站的 key  
    pay2go.hash_key    = "xxxxxxxx"  
    pay2go.hash_iv     = "xxxxxxxx"  
    pay2go.service_url = "https://api.pay2go.com/MPG/mpg_gateway"  
  end  
end
```

密码放在 .yaml 里面

- ▶ database.yaml
- ▶ s3.yaml (给同事知道有这个档)
- ▶ 然后 github 上传 database.yaml.example
- ▶ .gitignore 设定不要上传 database.yaml

坏人等着偷你的钥匙

Dev put AWS keys on Github. Then BAD THINGS happened ...

www.theregister.co.uk/.../dev_blunder_shows_github_cra... ▼ 翻譯這個網頁

2015年1月6日 - Fertile fields for **Bitcoin** yields - with a nasty financial sting ... to secure Rails apps which published his **Amazon** S3 keys to his **GitHub** account. ... but that was enough for a bot to pounce and spin up instances for **Bitcoin mining**.

There are now bots searching github for AWS keys ... - Reddit

https://www.reddit.com/.../Bitcoin/.../there_are_now_bots_... ▼ 翻譯這個網頁

2014年12月31日 - There are now bots searching **github** for AWS keys, then using them to **mine bitcoins** - My \$2375 **Amazon** EC2 Mistake (Not OP) (devfactor.net).

Developers, Check Your Amazon Bills For Bitcoin Miners ...

readwrite.com/.../amazon-web-services-hack-bitcoin-miner... ▼ 翻譯這個網頁

2014年4月15日 - **Bitcoin mining** was a common goal of these hackers, though the AWS computing ... After checking his **GitHub** repository, he found that he had ...

[新聞](#)
[產品評測](#)
[技術](#)
[專題](#)
[Big Data](#)
[Cloud](#)
[DevOps](#)
[資安](#)
[Video](#)
[研討會](#)
[社群](#)

新聞

GitHub擴充臭蟲害開發人員金鑰外洩10分鐘，AWS帳號馬上被盜用6500美元！

他的HumanCode儲存區只曝光了10分鐘，就已被駭客利用。Wyk推測這是因為許多比特幣的採礦者利用工具在網路上掃描AWS的存取金鑰，於AWS上建立大量的EC2實例以用來開採比特幣。

文/ 陳曉莉 | 2015-09-02 發表

✓ 讚
2.7 萬
按讚加入iThome粉絲團
f 讚
分享
663
G+
36

Billing & Cost Management Dashboard

Spend Summary

Indicate by the AWS Account Billing controls. Your last month's spend is \$6,485.39. Last month's spend is \$6,485.39. Current month's spend is \$6,485.39.

Month-to-Date Spend by Service

The chart below shows the proportion of costs spent for each service this year.

Month-to-Date Top Services by Spend

Service	Amount
EC2	\$6,485.39
S3	\$1.18
Data Transfer	\$1.07
CloudWatch	\$0.51
Other Services	\$1.00
Tax	\$0.00
Total	\$6,485.39

Alerts & Notifications

- Your account is enabled for automatic payment charges. Service fees being used to create an account which charges credit & debit card info.
- Also access to your account's billing information is not enabled. You can...

按讚追蹤 iThome 最新報導

✓ 讚
2.7 萬

Kanban精實軟體開發實作班

資策會「Kanban精實軟體開發實作班」(台北6/25)

\$ 7200 元起

正确观念

安全观念

- 密码用 figaro 管理
- ENV 环境变数