Ruby on Rails 購物車實作

使用歐付寶（卷二）

by @tonytonyjan

分離前後台 layout、controller

分離前後台 layout、controller

- 一般前後台不會共用 application.{js,css}、application.html.erb
- 有自己的 admin.{js,css}、admin.html.erb
- 所有的 admin namespace 下的 controller 繼承自 AdminController

分離前後台 layout、controller

執行

```
rails g assets admin
rails g controller admin
```

```ruby
# app/controllers/admin_controller.rb
class AdminController < ApplicationController
  layout 'admin'
end
# app/controllers/admin/orders_controller.rb
class Admin::OrdersController < AdminController
# app/controllers/admin/products_controller.rb
class Admin::ProductsController < AdminController
# config/initializers/assets.rb
Rails.application.config.assets.precompile += %w(admin.js admin.css)
```

將 `application.{css,js}` 中的 `require_tree .` 拿掉。

後台版型

```
<!-- app/views/layouts/admin.html.erb -->
<!DOCTYPE html>
<html>
<head>
  <title>MyCart Admin</title>
  <%= stylesheet_link_tag    'admin', media: 'all', 'data-turbolinks-track' => true %>
  <%= javascript_include_tag 'admin', 'data-turbolinks-track' => true %>
  <%= csrf_meta_tags %>
</head>
<body>

<%= yield %>

</body>
</html>
```

練習

加上後台的導覽列

後台帳號密碼

```ruby
# app/controllers/admin_controller.rb
class AdminController < ApplicationController
  if Rails.env.production?
    http_basic_authenticate_with(
      name: 'tonytonyjan', password: 'secret'
    )
  end
  layout 'admin'
end
```

訂單與交易

歐付寶金流文件

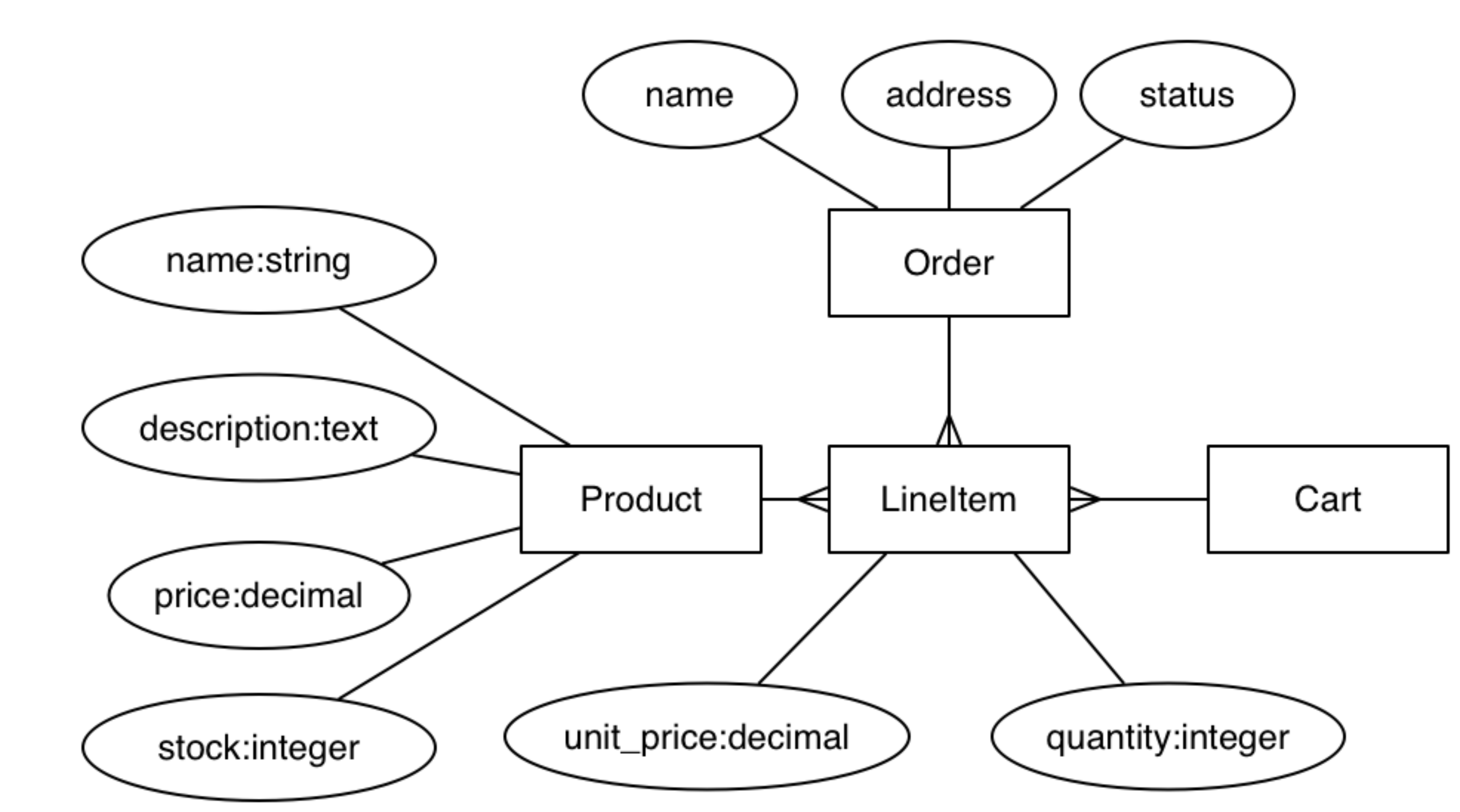
交易順序

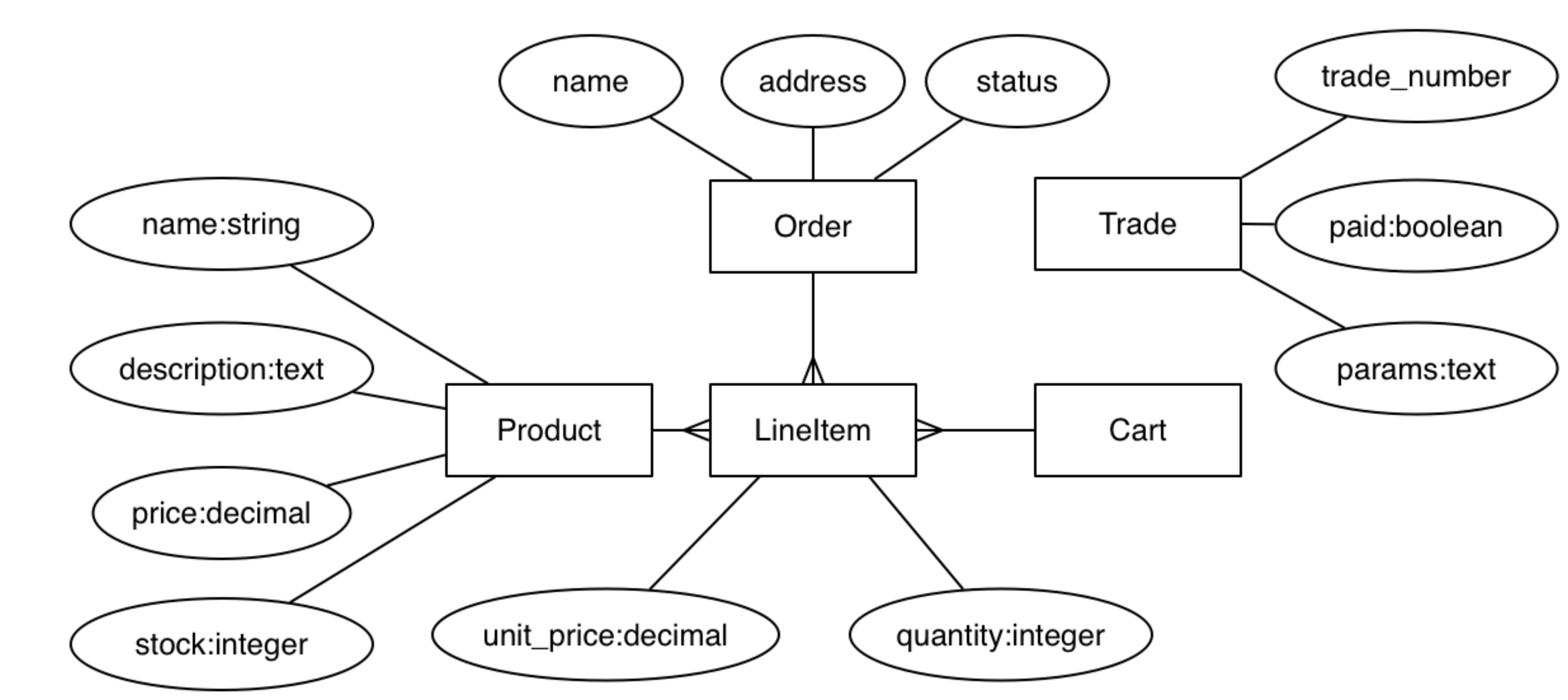
1. 網站產生訂單交易編號

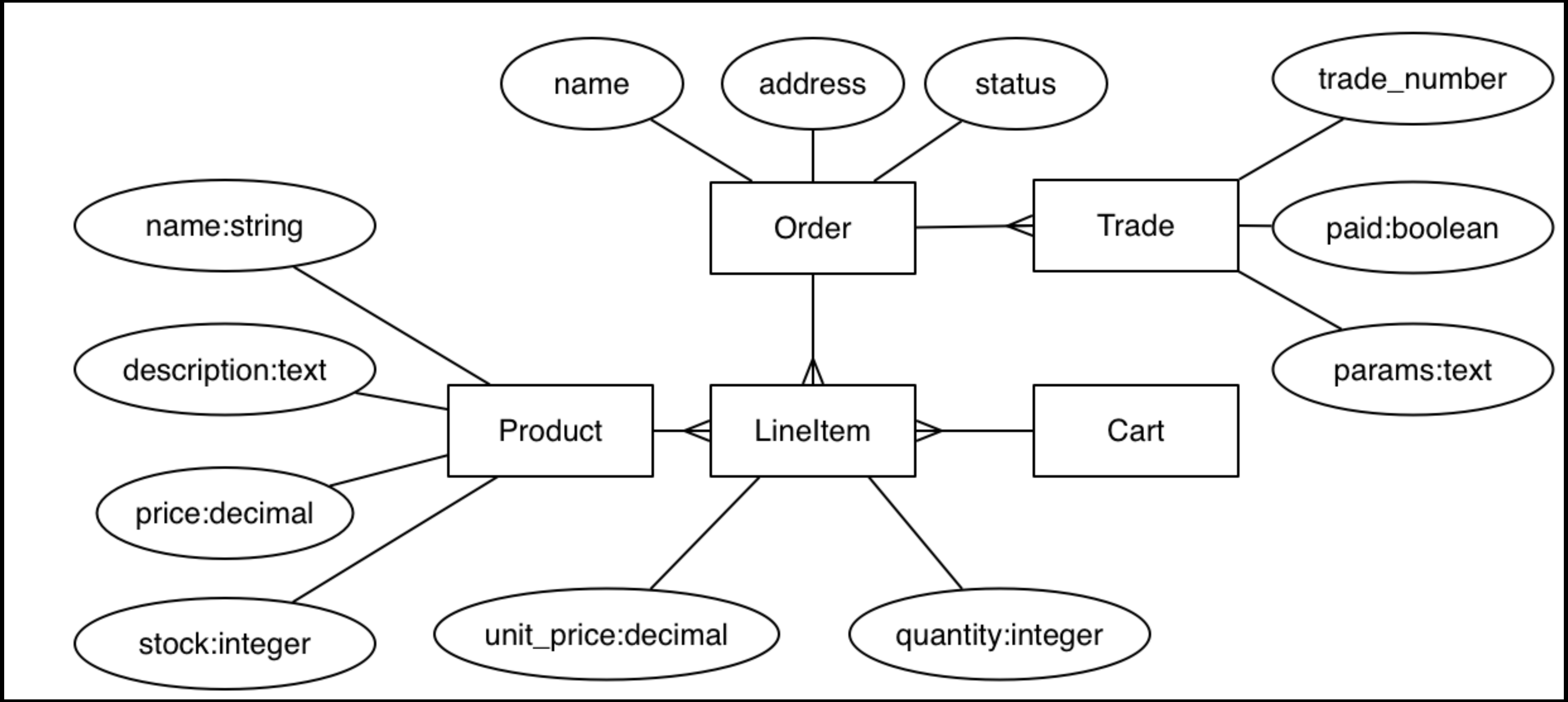
2. 網站產生以此交易編號表單 POST 至歐付寶

3. 使用者跳轉至歐付寶進行付款

4. 歐付寶以 POST 夾帶交易編號與交易結果回訪網站


name
address
status
Order
name:string
description:text
Product
LineItem
Cart
price:decimal
stock:integer
unit_price:decimal
quantity:integer


name
address
status
trade_number
Order
Trade
name:string
paid:boolean
description:text
params:text
Product
LineItem
Cart
price:decimal
stock:integer
unit_price:decimal
quantity:integer


name
address
status
trade_number
Order
Trade
name:string
paid:boolean
description:text
params:text
Product
LineItem
Cart
price:decimal
stock:integer
unit_price:decimal
quantity:integer

產生交易紀錄

```
rails g model Trade order:references \
trade_number:index paid:boolean params:text
```

```
# db/migrate/VERSION_create_trades.rb
t.references :order, index: true, foreign_key: true, null: false
t.string :trade_number, null: false
t.boolean :paid, null: false, default: false
# app/models/order.rb
has_many :trades
```

trade.rb

```ruby
# app/models/trade.rb

# 讓 params 欄位可以存 hash
serialize :params, JSON

# 初始化 Trade 後產生交易編號
after_initialize :generate_trade_number, if: :new_record?
def generate_trade_number
  self.trade_number = SecureRandom.hex(3)
end
```

rake dev:fakeup

```
paid_order = Order.create name: 'Tony', address: '樹林', payment_method: 'Credit'
paid_order.line_items.create(
  product: Product.all.sample, unit_price: 123, quantity: 1
)
paid_order.trades.create paid: true

unpaid_order = Order.create name: 'Jason', address: '板橋', payment_method: 'ATM'
unpaid_order.line_items.create(
  product: Product.all.sample, unit_price: 456, quantity: 2
)
unpaid_order.trades.create paid: false
```

練習

```
rails g controller orders show new checkout
```

- Order#paid?
- /orders/:id
- /orders/new
- 在 /cart 加上 new*order*path 連結
- 在後台呈現 Order 付款狀態

新增訂單

```ruby
# config/routes.rb
resources :orders, only: %i[show new create] do
  get :checkout
end

# app/controllers/orders_controller.rb
def create
  @order = Order.new_from_cart(current_cart, order_params)
  if @order.save
    current_cart.clear
    redirect_to @order
  else
    render :new
  end
end
```

練習

- Order#new*from*cart(cart, params)

根據 params 初始化，並將 cart 中的 items 搬家。

扣除庫存時機點

1. 加入購物車後
2. 建立訂單後
3. 交易成功後

建立訂單後扣除庫存

```ruby
# app/models/order.rb
after_create :consume_stock
private
def consume_stock
  line_items.each do |line_item|
    line_item.product.decrement! :stock, line_item.quantity
  end
end
```

歐付寶訂單最小範例

```
<form action="http://payment-stage.allpay.com.tw/Cashier/AioCheckOut" method="post">
    <input type="text" name="MerchantID" value="2000132" >
    <input type="text" name="MerchantTradeNo" value="3baf9a" >
    <input type="text" name="MerchantTradeDate" value="2015/05/05 07:55:06" >
    <input type="text" name="PaymentType" value="aio" >
    <input type="text" name="TotalAmount" value="1500" >
    <input type="text" name="TradeDesc" value="腦袋有動雜貨店" >
    <input type="text" name="ItemName" value="項目一#項目二" >
    <input type="text" name="ChoosePayment" value="Credit" >
    <input type="text" name="ReturnURL" value="http://host/allpay/return" >
    <input type="text" name="OrderResultURL" value="http://host/allpay/result">
    <input type="text" name="ClientBackURL" value="http://host/orders/766573115" >
    <input type="text" name="CheckMacValue" value="0EB8909F1B24749D50BD5A0C8D5370EB" >
  <input type="submit">
</form>
```

重要參數說明

- ReturnURL - 交易結果回傳網址
- OrderResultURL - 不顯示歐付寶交易結果頁面，直接回訪
- ClientBackURL - 交易後的返回商店網址
- CheckMacValue - 檢查碼，是由所有的參數算出來的

歐付寶測試資料

```
Attribute    | Value
------------ | -------------------
MerchantID   | 2000132
HashKey      | 5294y06JbISpM5x9
HashIV       | v77hoKGq4kWxNNIS
Credit Cartd | 4311-9522-2222-2222
CVV          | 222
```

View 應只負責顯示

```
<!-- app/views/orders/checkout.html.erb -->
<h1>處理中……</h1>
<form action="http://payment-stage.allpay.com.tw/Cashier/AioCheckOut"
      method="post"
      style="display:none">
  <% @checkout_params.each do |name, value| %>
    <%= hidden_field_tag name, value %>
  <% end %>
  <input type="submit">
</form>
<script>document.forms[0].submit()</script>
```

實作 Allpay 檢查碼

檢查碼規則

將傳遞參數依照第一個英文字母，由 A 到 Z 的順序來排序(遇到第一個英名字母相同時，以第二個英名字母來比較，以此類推)，並且以&方式將所有參數串連後，再將參數最前面加上 HashKey、最後面加上 HashIV，並將整串字串進行 URL encode 後再轉為小寫，再以 MD5 加密方式來產生 CheckMacValue。

— 節錄自歐付寶文件

白話版本

1. 將參數依字典排序
2. 以 `key=value` 的方式呈現，用 & 連接
3. 首尾個別加上 `HashKey`、`HashIV`
4. URL 編碼後轉小寫
5. MD5 加密後轉大寫

練習

```ruby
# lib/allpay.rb
class Allpay
  attr_accessor :merchant_id, :hash_key, :hash_iv

  def initialize merchant_id: '2000132', hash_key: '5294y06JbISpM5x9', hash_iv: 'v77hoKGq4kWxNNIS'
    @merchant_id = merchant_id
    @hash_key = hash_key
    @hash_iv = hash_iv
  end

  def make_mac params = {}
    # Hash#sort
    # Array#join
    # CGI#escape
    # Diges::MD5#hexdigest
    # String#downcase
    # String#upcase
  end
end
```

```ruby
# config/routes.rb
post 'allpay/result'
post 'allpay/return'
```

```ruby
# app/controllers/orders_controller.rb
def checkout
  order = Order.find params[:order_id]
  trade = order.trades.create!
  allpay = Allpay.new
  @checkout_params = {
    MerchantID: allpay.merchant_id,
    MerchantTradeNo: trade.trade_number,
    MerchantTradeDate: Time.now.strftime('%Y/%m/%d %H:%M:%S'),
    PaymentType: :aio,
    TotalAmount: order.total.round,
    TradeDesc: :'My Cart',
    ItemName: order.line_items.map{ |i| "#{i.product.name} x #{i.quantity}" }.join('#'),
    ChoosePayment: order.payment_method,
    ReturnURL: 'http://requestb.in/r9vwt1r9',
    OrderResultURL: allpay_result_url
  }
  @checkout_params[:CheckMacValue] = allpay.make_mac(@checkout_params)
end
```

紀錄交易紀錄

```
rails g controller allpay
```

```
# app/controllers/allpay_controller.rb
require 'allpay'
class AllpayController < ApplicationController
  skip_before_action :verify_authenticity_token
  before_action :set_trade

  def result
    if @trade.save
      redirect_to @trade.order, alert: '交易成功'
    else
      redirect_to @trade.order, alert: '交易失敗'
    end
  end

private

  def set_trade
    @trade = Trade.find_and_initialize_by_allpay(request.POST)
  end
end
```

歐付寶回訪參數

```
RtnMsg: 交易成功
CheckMacValue: B740F880764475931B1CC23250DFAA56
SimulatePaid: 0
TradeDate: 2015/05/05 09:48:54
PaymentTypeChargeFee: 30
PaymentDate: 2015/05/05 09:49:24
MerchantID: 2000132
TradeAmt: 1500
RtnCode: 1
TradeNo: 1505050948543364
PaymentType: Credit_CreditCard
MerchantTradeNo: 09a4d7
```

實作 Trade.find*and*initialize*by*allpay

```ruby
# app/models/trade.rb
def self.find_and_initialize_by_allpay params = {}
  trade = self.find_by(trade_number: params[:MerchantTradeNo])
  trade.paid = params['RtnCode'] == '1'
  trade.params = params
  trade
end
```

紀錄交易（透過歐付寶回訪）

```ruby
# app/controllers/allpay_controller.rb
def return
  if @trade.save
    render text: :'1|OK'
  else
    render text: :'0|ErrorMessage'
  end
end
```

檢查碼驗證

```ruby
# lib/allpay.rb
def check_mac params = {}
  stringified = params.stringify_keys
  check_mac_value = stringified.delete('CheckMacValue')
  make_mac(stringified) == check_mac_value
end
```

練習

- 在 trade.rb 加上檢查碼 validation
- 防止重複付款