



招聘系統

主要機制

- ▶ 張貼系統 - 開缺公司，撰寫工作需求
- ▶ 觀看機制 - 應徵者，過濾理想工作
- ▶ 應徵系統 - 應徵者，可以上傳簡歷
- ▶ 管理系統 - 管理員可以刪除隱藏工作

Title Goes Here

- ▶ 張貼系統 - 開缺公司, 撰寫工作需求
- ▶ 觀看機制 - 應徵者, 過濾理想工作
- ▶ 應徵系統 - 應徵者, 可以上傳簡歷
- ▶ 管理系統 - 管理員可以刪除隱藏工作

Title Goes Here

- ▶ 開缺公司
- ▶ 應徵者
- ▶ 管理員

Title Goes Here

- ▶ 開缺公司 = 管理員
- ▶ 應徵者

- ▶ 張貼系統 - 管理員，撰寫需求
- ▶ 管理系統 - 管理員可以刪除隱藏工作
- ▶ 觀看機制 - 應徵者，過濾理想工作
- ▶ 應徵系統 - 應徵者，可以上傳簡歷

接下來，我們要來介紹一個新東西

User Story

用戶故事

User Story (用戶故事)

- ▶ 通常用在敏捷項目之中
- ▶ 釐清「會發生的故事」，而不是「規格」
- ▶ 以角色為中心點

格式

- 身為「某角色」，會做「某事」，以完成「某商業價值」

- 身為管理員，可以張貼職缺，以達到招募人才的需求
- 身為應徵者，可以篩選職缺，以達到「篩選到自己心儀薪資水準的工作」
- ...
- ...
- ...
- 小型專案約20-30條。中型約100條。

為什麼要有 User Story (更貼近真實)

- › 客戶說：系統要有訂單系統
- › 快結案了卻只寫了前台的客戶的訂單系統
- › 發案的人說：後台需要出貨、對帳、物流...
- › feature 暴增 10 倍

常見專案失敗之處

- › 顧著開發「規格上的東西」，然後 90% 時...

常見專案失敗之處

- › 顧著開發「規格上的東西」，然後 90% 時...
- › 少了「重要角色」

常見專案失敗之處

- › 顧著開發「規格上的東西」，然後 90% 時...
- › 少了「重要角色」
- › 寫太細，新手 Developer 不知道怎麼開發

項目示範

VI

- ▶ 張貼系統 - 管理員，撰寫需求
- ▶ 管理系統 - 管理員，可以刪除隱藏工作
- ▶ 觀看機制 - 應徵者，過濾理想工作
- ▶ 應徵系統 - 應徵者，可以上傳簡歷

太多東西了，不知道怎麼下手....

V1

- ▶ 張貼系統 - 管理員, 撰寫需求
- ▶ 管理系統 - 管理員, 可以刪除隱藏工作
- ▶ 觀看機制 - 應徵者, 過濾理想工作
- ▶ 應徵系統 - 應徵者, 可以上傳簡歷

角色分兩種

- ▶ 管理員
- ▶ 應徵者

角色分兩種

- ▶ 管理員
- ▶ 應徵者

V2

- › 身為管理者， 我可以在新增職缺需求
- › 身為管理者， 我可以編輯 / 刪除職缺需求
- › 身為管理者， 我可以隱藏職缺需求

在哪裡做呢？

V2

- 身為管理者，我可以在「後台」新增職缺需求
- 身為管理者，我可以在「後台」編輯 / 刪除職缺需求
- 身為管理者，我可以在「後台」隱藏職缺需求

還有什麼是要深究的？

V2

- › 身為管理者， 我可以在新增職缺需求
- › 身為管理者， 我可以編輯 / 刪除職缺需求
- › 身為管理者， 我可以隱藏職缺需求

職缺的「最小資訊量」

V3

- 必須要有「後台」、「管理員」、「登入的設計」
- 欄位必須要有「標題」 「內容」 「薪資上限」 「薪資下限」 「聯繫方式」
- 欄位必須要有「是否隱藏」

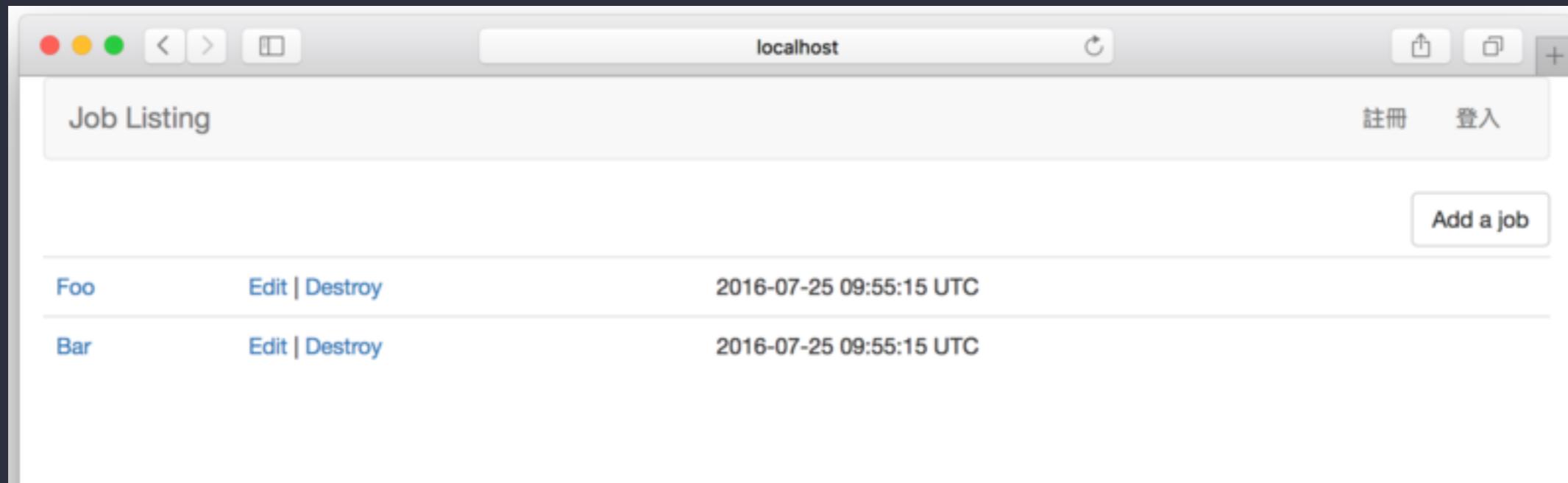
V4 第一個實作版本

- 身為管理者， 我可以在新增職缺需求
 - 包括「標題」 「內容」 「薪資上限」 「薪資下限」 「聯繫方式」
- 身為管理者， 我可以編輯 / 刪除職缺需求 （在後台）
- 身為管理者， 我可以隱藏職缺需求 （在後台）
- 身為使用者， 我必須「登入」 才能進發表職缺
- 身為使用者， 我必須要有「admin 身份註記」 才能進入後台

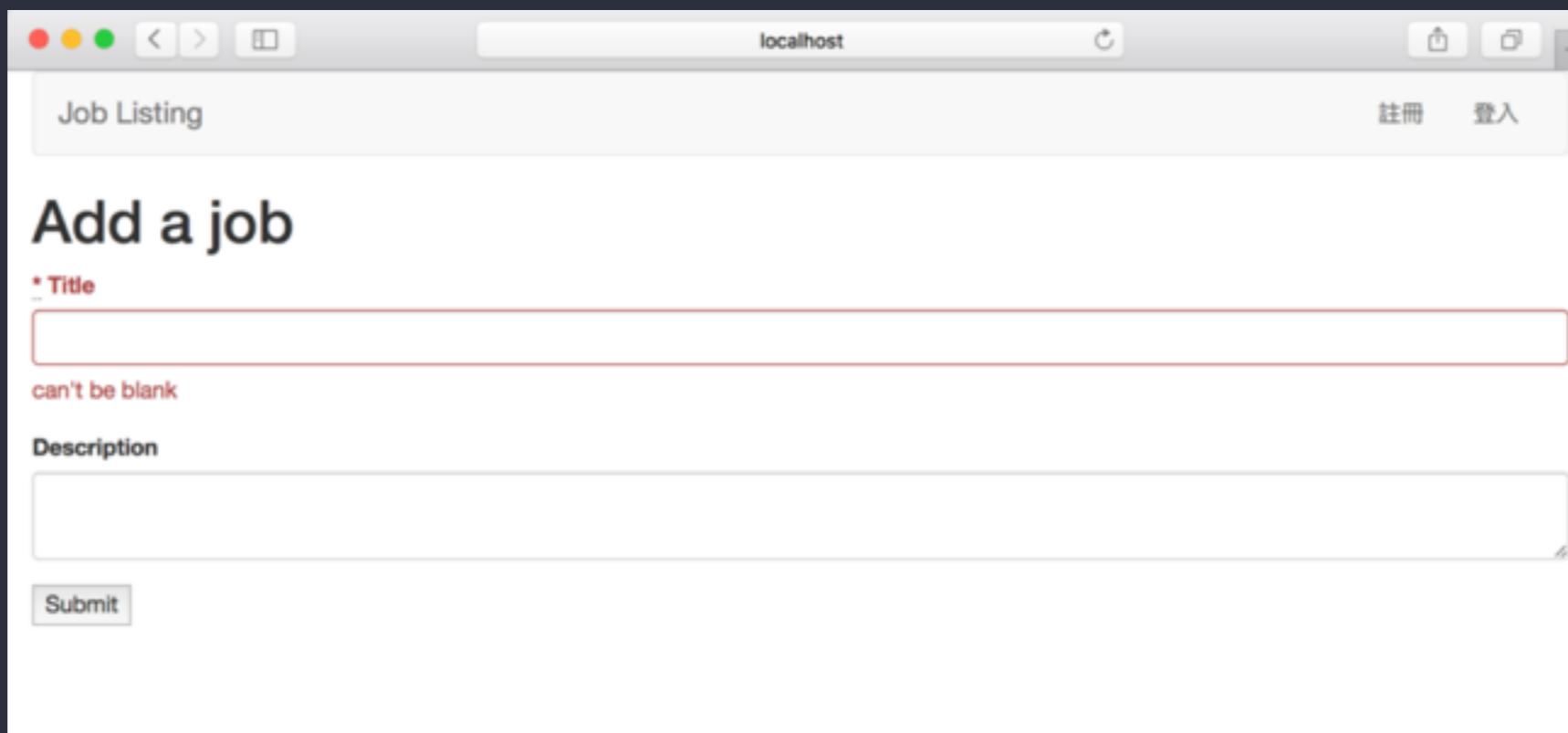
實作步驟

Step 1

- › 實作 Job 的 CRUD (JobsController)
- › 先只開 Job 的 title, description
- › (最小可行性介面)



Job Listing		
Foo	Edit Destroy	2016-07-25 09:55:15 UTC
Bar	Edit Destroy	2016-07-25 09:55:15 UTC



Add a job

* Title
 can't be blank

Description

Step 1

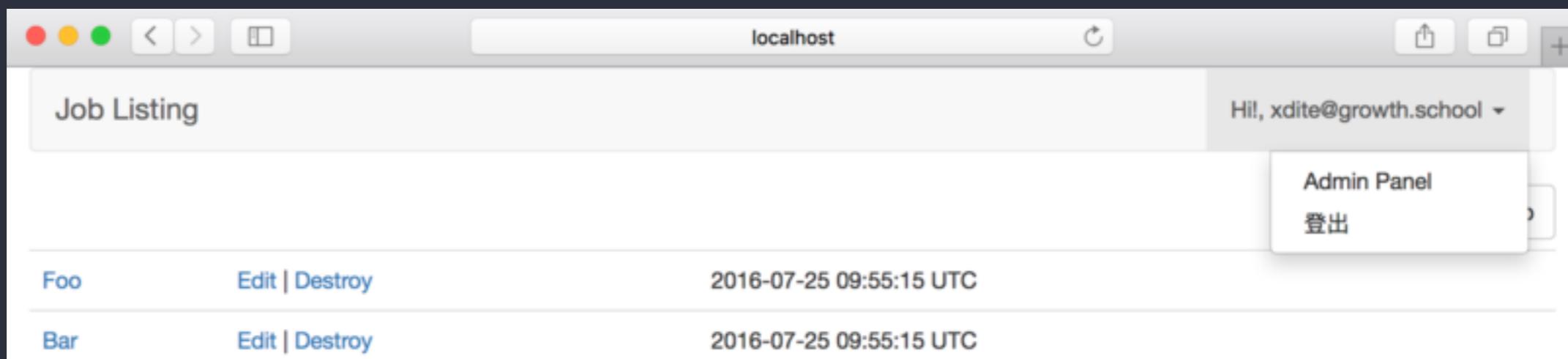
- 要新增 / 修改 / 刪除 Job 必須得登入

```
app/controllers/jobs_controller.rb
```

```
before_filter :authenticate_user!, only: [:new, :create, :update, :edit, :destroy]
```

Step 2

- ▶ 實作 Job 的 Admin 的 CRUD, 在 admin/jobs
- ▶ 下拉選單要有一個連結可以到 admin_jobs_path



Step 2

- ▶ 在 admin / jobs 皆需要先過 require_is_admin 才能進入，否則回去首頁
- ▶ 定義一個條件 admin? 判斷是否是 admin

```
class Admin::JobsController < ApplicationController
  before_filter :authenticate_user!, only: [:new, :create, :update, :edit, :destroy]
+ before_filter :require_is_admin
```

app/controller/jobs_controller.rb

```
def require_is_admin
  if !current_user.admin?
    flash[:alert] = 'You are not admin'
    redirect_to root_path
  end
end
```

app/models/user.rb

```
def admin?
  is_admin
end
```

Step 3

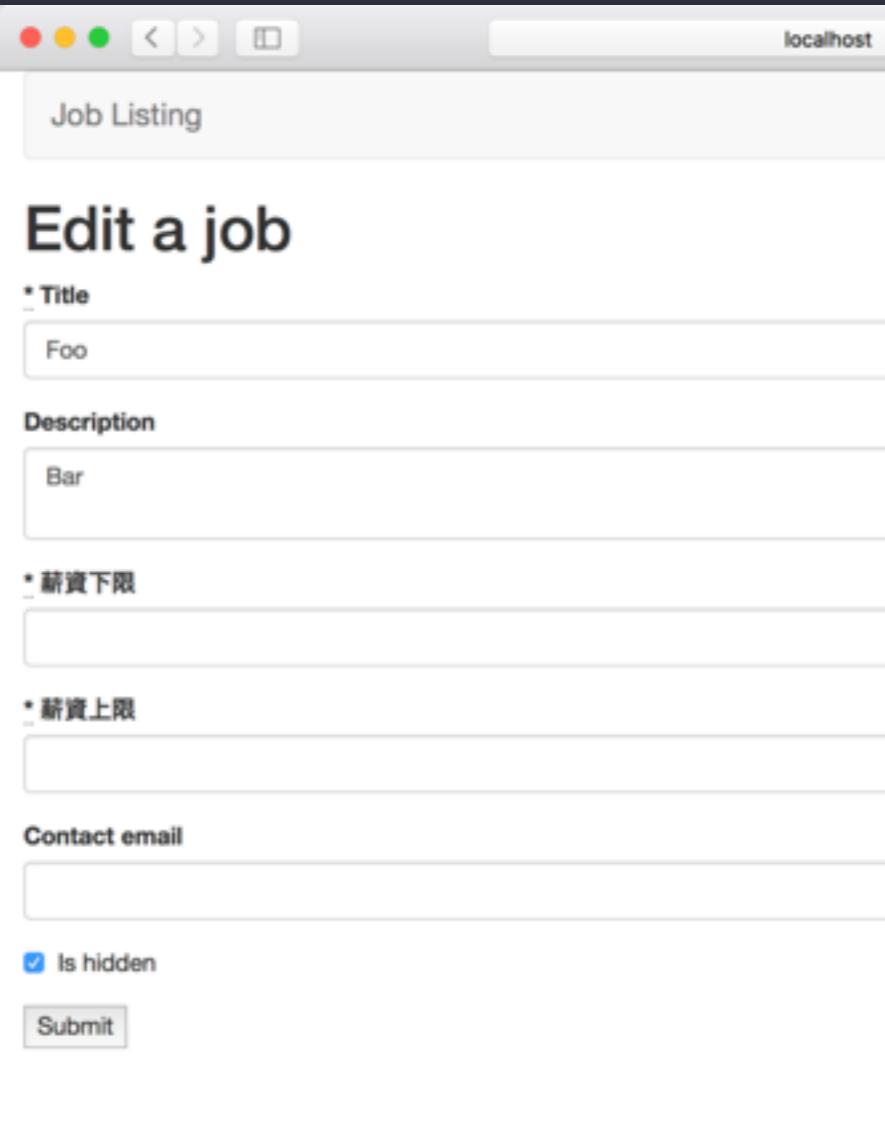
- ▶ 新增「薪資上限」與「薪資下限」、「聯絡方式」
- ▶ wage_upper_bound, wage_lower_bound,
contact_email
- ▶ 「薪資上限」、「薪資下限」不得為空
- ▶ 「薪資」必須是數字

```
app/models/job.rb
```

```
validates :wage_upper_bound, presence: true
validates :wage_lower_bound, presence: true
validates :wage_lower_bound, numericality: { greater_than: 0}
```

Step 4

- ▶ 管理者可以設定職缺是否在前臺隱藏



Job Listing

Edit a job

* Title
Foo

Description
Bar

* 薪資下限

* 薪資上限

Contact email

Is hidden

Submit

Step 4

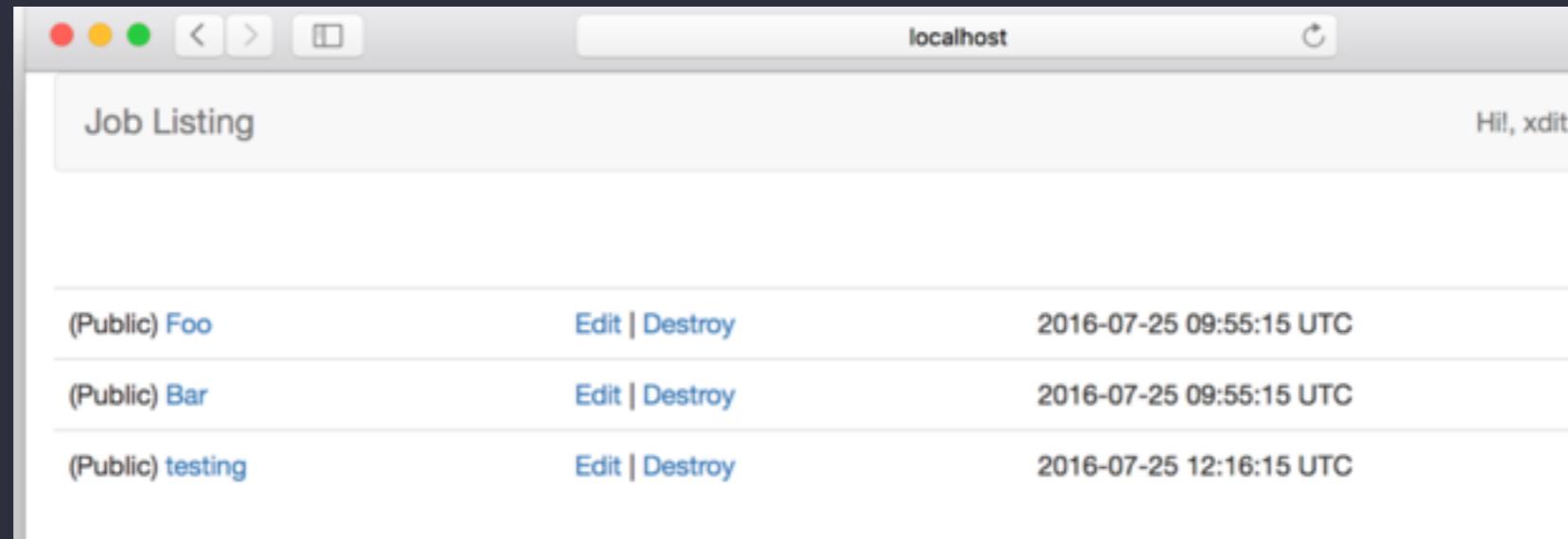
- ▶ 被設為隱藏的職缺，是不可以在前台看到的。

```
def index
  @jobs = Job.where(:is_hidden => false)
end
```

Step 4

- ▶ 首頁文章應該按照發表時間順序

```
def index
  @jobs = Job.where(:is_hidden => false).order("created_at DESC")
end
```



Job Listing		
(Public) Foo	Edit Destroy	2016-07-25 09:55:15 UTC
(Public) Bar	Edit Destroy	2016-07-25 09:55:15 UTC
(Public) testing	Edit Destroy	2016-07-25 12:16:15 UTC

總結

- ▶ before_action (過濾器)
- ▶ validates (驗證條件)
- ▶ where (拿來判斷條件)
- ▶ order (拿來排序)
- ▶ Helper (View 裡面寫 Ruby 邏輯的工具)