

**Rails 第三课: Rails 101 (/courses/3/syllabus)** 🔍 搜索本课程教材

📖 概览 (/courses/3) — 📖 教材 (/courses/3/syllabus)

📁 作业 (/courses/3/assignments)

💬 FAQ (/courses/3/faqs) 📡 动态 (/courses/3/activities)

☆ 积分榜 (/courses/3/leaderboard)

有 0 位同学正在浏览当前页面

## 4-9 (解说) Rails 的 CRUD 与 RESTful

🔖 (/favorites?post\_id=69)

如果你还在做第一遍练习时, 请跳过这整章解说, 否则你大脑会乱掉!

如果你还在做第一遍练习时, 请跳过这整章解说, 否则你大脑会乱掉!

如果你还在做第一遍练习时, 请跳过这整章解说, 否则你大脑会乱掉!

- 叫你滚出去还不明白吗?
- 叫你滚出去还不明白吗?
- 叫你滚出去还不明白吗?

好。我现在确定你应该已经是做第二遍了, 才看这一篇解说的。

## HTTP 请求( Request) 与 HTTP 动作 ( Verb)

一般网页的运作原理是这样的:

1. 使用者在浏览器输入了某个网址, 点击了某个连结。这都叫发出 HTTP 请求 ( Request )。
2. 而浏览器回应之后: 产生页面, 或者是重新导向。这都叫 HTTP 回应 ( Response )。

一般来说, 使用者最常使用的是两种类型 / 动作 ( Verb )的请求: GET 与 POST。

- GET: 输入网址, 点击某些网址。这都是 GET。
- POST: 送出表单。这叫做 POST

## RESTful 的基础概念

因为 CRUD 是常见的操作行为, 但是大家实作的方式、与网址设计方式不一样。造成很多维护上的困扰。

后来就有人发明 RESTful 这个概念, 希望用 HTTP 的动作 ( Verb )一起去封装 CRUD 的行为。所以又多加了两个动作:

- PUT: 更新 ( update )
- DELETE: 删除 ( delete )

而 Rails 的 Routing 也是依循这样的理念去设计。

GET	POST	PUT/PATCH	DELETE
讀取	新增	更新	刪除
Read	Create	Update	Destroy

(https://s3-ap-northeast-

1.amazonaws.com/ontrackapp-production/rGUHz08gQriXQbiPLTE4\_%E8%9E%A2%E5%B9%95%E5%BF%AB%E7%85%A7%202016-07-13%20%E4%B8%8A%E5%8D%881.29.48.png)

## Rails 的基础 RESTful概念

由于RESTful是个新手不容易一下子理解的概念，号称学 Ruby on Rails 初期最大障碍。

在本篇尽量用浅显易懂的方式解释。如果看不懂也没关系，用背的！！

一定要背起来！

Helpers  
(以Rouces: group為例)

HTTP Verb  
(HTTP的request動詞)

group\_s\_path

group\_path(group)

edit\_group\_path(group)

new\_group\_path

以上輔助函式(helper)  
放在Views裡就能  
轉成正確的連結網址

Controllers  
(輸出後的網址)

HTTP Verb  
(HTTP的request動詞)

index action  
( http://your\_project\_name.com/groups )

show action  
( http://your\_project\_name.com/groups/[:id] )

edit action  
( http://your\_project\_name.com/groups/[:id]/edit )

new action  
( http://your\_project\_name.com/groups/new )

所有GET的action都有對應  
各自專屬的view

(https://s3-ap-

northeast-1.amazonaws.com/ontrackapp-production/WJXUF3ISuyzR26tmyNc9\_xDdK3FnmSXYZC9mUxjtL\_TPgJPeyuQkyNumT3bJGA\_RESTful.jpg)

## 以下例子以 第三章， 做出 CRUD 功能的讨论版来解释：

---

- 讨论版这个功能的名称： `groups`。Restful 的概念里面叫做 `resources`，“一组资源”。

为了要控制它的运作，我们会创建一个新的 Controller文件 叫做 `groups_controller.rb`

HTTP Verb 有四种：

- 新增 (`create`，使用 `POST`)
- 读取 (`read`，使用 `GET`)
- 更新 (`update`，使用 `PUT`)
- 删除 (`destroy`，使用 `DELETE`)

在这组 `controller` 当中，我们会实作 7 个 `Action`，4 张页面。

- 首页(`index`)：用来列出所有的讨论版，可以选择各个单版（使用 `GET`）
- 各个讨论版专属页面(`show`)：显示讨论版版名跟简介 (使用 `GET`)
- 新增页面(`new`)：里面会有表单，填完以后可以送出 (使用 `GET`)
- 产生数据 (`create`)： `new` 送出来的表单到 `create` 这个 `action`，新增一笔数据 (使用 `POST`)
- 编辑页面(`edit`)：里面会有表单呈现现有数据来，填完数据后可以送出 (使用 `GET`)
- 更新数据 (`update`)： `edit` 送出来的表单可以到 `update` 这个 `action`，更新该笔数据 (使用 `PUT`)
- 删除数据：送出删除请求，删除该笔数据 (使用 `DELETE`)

这七个 `action`，是每个CRUD功能里最基本的动作。只要我们在 `controller.rb` 里定义 (`def`) 好这七个方法当作动作 (`action`)。其他的部分 Rails 就会帮我们处理了。

像是设定 Routing (路由) 只需要简单一行：

```
config/routes.rb
resources :groups
```

这样不管浏览器端跑来什么样的请求 (`request`)，Routing都会聪明地帮我们导向正确的 `action` 去运作后续流程。

## 网址对应关系

---

对于网址跟对应的Action与页面来说

- `/groups/` 一定就是 `index`，首页，对应Routes路径： `groups_path`
- `/groups/123` 一定就是 `show`，名字为123的讨论版个版页面，对应Routes路径： `group_path(123)`
- `/groups/123/edit` 一定就是 `edit`，编辑讨论版123的表单页面，对应Routes路径：  
`edit_group_path(123)`
- `/groups/new` 一定就是 `new`，新增讨论版的表单页面，对应Routes路径(或称为helper):  
`new_group_path` 总结

所以制作一个 RESTful 风格的讨论版，我们仅需在Routes设定一行代码即可

对本页内容的感受如何？

我要吐槽

😊 So easy

😐 还OK

😫 崩溃了

← 上一页 (/posts/68)

🔍 可以使用 ← → 键进行翻页

下一页 ➡ (/posts/70)

- 全栈营

课程介绍  
(/pages/course\_intro)

教学团队  
(/pages/teachers)

学员心得  
(/pages/students)
- 课程资源

学习中心  
(/dashboard)

帮助文档  
(http://docs.qzy.camp/)


交流论坛  
(http://forum.qzy.camp/)
- 关于我们

公司介绍  
(/pages/about)

常见问题  
(/pages/faq)

联系方式  
(/pages/contact)

在线客服（非技术答疑，工作日10:00-19:00）



新生大学 - 软件学院是李笑来对未来世界的实验计划，旨在改变中国的计算机教育。  
(http://www.xinshengdaxue.com/)