

全栈营 (/dashboard)

学习中心 (/dashboard) Meetup (/meetup_groups) 直播室 (/live) 编程比赛 (/competition_season_2)

帮助文档 (<http://docs.qzy.camp/>) 交流论坛 (<http://forum.qzy.camp/>)

✉ (/conversations) 👤 江帆 ▾

Rails 第二课: 初级练习 (/courses/2/syllabus) 🔍 搜索本课程教材

📖 概览 (/courses/2) ≡ 教材 (/courses/2/syllabus)

📁 作业 (/courses/2/assignments)

💬 FAQ (/courses/2/faqs) 📡 动态 (/courses/2/activities)

☆ 积分榜 (/courses/2/leaderboard)

有 0 位同学正在浏览当前页面

4-1 补充教材: Rails 架构

🔖 (/favorites?post_id=49)

目标

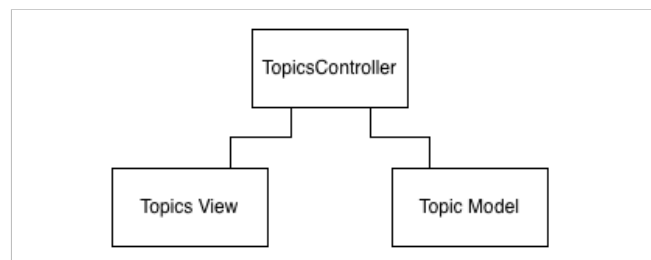
做一个资料库表格来储存 topic, topic 有 title 和 description。

在这一步我们会学一点 Rails 的架构。最终你应该会了解这些观念:

- Table
- Model
- View
- Controller

解说

Rails 的架构以及它与资料库的关联



([https://s3-ap-northeast-1.amazonaws.com/ontrackapp-](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/2aFSZZRQKei3eSroobFg_%E8%9E%A2%E5%B9%95%E5%BF%AB%E7%85%A7%202016-07-10%20%E4%B8%8B%E5%8D%8811.14.22.png)

[production/2aFSZZRQKei3eSroobFg_%E8%9E%A2%E5%B9%95%E5%BF%AB%E7%85%A7%202016-07-10%20%E4%B8%8B%E5%8D%8811.14.22.png](https://s3-ap-northeast-1.amazonaws.com/ontrackapp-production/2aFSZZRQKei3eSroobFg_%E8%9E%A2%E5%B9%95%E5%BF%AB%E7%85%A7%202016-07-10%20%E4%B8%8B%E5%8D%8811.14.22.png))

Rails 把 **Model/View/Controller** 的设计模式实作得非常具体，来指引你如何设计你的网路应用程式。

Model

- 我们在 RailsBridge 建立的 Model，其每一个 Model object 都在资料库里面有对应的资料。资料库里面的表格 (table) 是 Model 的 class name 的复数形。例如，如果 Model 叫做 'Duck'，就会自动去资料库读写 'ducks' 这个表格。
- Rails 内部的 methods 让我们可以很容易把资料写入到资料库，之后再从资料库里面查询 (query) 出来。
- Model 是介于资料库和你的程式码之间的桥梁。

View

- View 会产生 HTML 来显示在浏览器。
- View 档案是用 ERB 写的，它是一种样板语言 (Template Language)。里面是 HTML 加上内嵌的 Ruby 程式码。View 里面的 Ruby 的变数便是当使用者要浏览该页面的时候，所要填入的内容。
- (还有别的样板语言，但是在 RailsBridge 我们只用 ERB。)

Controller

- Controller 把 Ruby 的 objects 在 Model 和 View 之间传来传去。
- 每一个 URL 都对应到 Controller 里面的某一个特定的 method。
- 在这一步骤以后，当你打开你应用程式里面的任何一个页面，该请求 (request) 会被某个 Controller 的 method 处理。

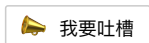
当我们把 Models、Views、Controllers 放在一起的时候，他们会遵循以下的模式：


给一个 URL，Rails 会去检查要使用哪一个 Controller 里面的 method (又称为 "Action")。Controller Action 会去呼叫 Model 里面对应的 methods。Model 会去读写资料库，然后把包含资料的 object 回传到 Controller。Controller 会拿到这个 object 并且丢到 View 里面。Action 通常会有对应的 View 档案，Rails 会自动寻找并使用之。)

Models、Views、Controllers 有各自的工作。像这样把责任拆开来，会比较容易开发，尤其是它长得愈来愈大的时候。(每个档案都有清晰的责任，会比较容易处理问题、增加新功能。)

如果你想要学更多 Rails 架构的话，你可以看一下这个教学影片 (三分半钟) “[MVC 架构 YouTube](https://www.youtube.com/watch?v=eTdVkgF_Slo)” (https://www.youtube.com/watch?v=eTdVkgF_Slo)

对本页内容的感受如何？



 So easy

 还OK

 崩溃了

← 上一页 (/posts/40)

💡 可以使用 ← → 键进行翻页

完成本课 (/dashboard)

全栈营	课程资源	关于我们	 在线客服 (非技术答疑, 工作日10:00-19:00)
课程介绍 (/pages/course_intro)(/dashboard)	学习中心	公司介绍 (/pages/about)	 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/)
教学团队 (/pages/teachers) (http://docs.qzy.camp/)	帮助文档	常见问题 (/pages/faq)	
学员心得	交流论坛	联系方式	
(/pages/students) (http://forum.qzy.camp/)		(/pages/contact)	