

2-2 建立新项目

- * 建立一个新 Rails 项目: `rails new rails101`;
- * 进入新建好的项目, 并建立 git 做版本控制;

2-3 帮 Rails 项目穿上衣服“Bootstrap”

- * 帮项目穿上衣服 Bootstrap, 挂上 bootstrap-sass 这个 gem
- * app/assets/stylesheets/application.scss:

...

```
@import "bootstrap-sprockets";  
@import "bootstrap";
```

...

- * 新增 app/views/common 文件夹
- * 新增 navbar, footer
- * 修改全域 HTML 样式 application.html.erb
- * 新增一个 welcome controller
- * 新增一个空的 Hello World 页面
- * 将首页指到 welcome 下的 index.html.erb 页面

2-5 制作漂亮的“提示信息”

- * 将 Bootstrap 的 js 提示套件 bootstrap/alert “挂”进项目里面
- * 加入 app/helpers/flash_helper.rb
- * 新增 app/views/common/_flash.html.erb
- * 在 application.html.erb 内加入 flash 这个 partial

3-1 建立讨论群的架构

- * 设计 Group 的 model 架构
- * 产生 groups controller
- * 建立 index action 以及 index 的 view
- * 在 routing 上挂上 groups

3-2 将首页换成讨论群一览表

- * 修改 routing

4-2 实作讨论群“新增”功能

- * new, create
- * show
- * edit, update, destroy

4-6 限制“标题为空”的Group, 不能被新增

- * 在 Group model 加入“标题限制”
- * new, update

4-7 共用“表单”

- * 产生 app/views/groups/_form.html.erb (edit, new)
- * "move form to partial"

4-8 将表单换为 Bootstrap 提供的版型

- * 使用 SimpleForm 简化

5-2 安装 devise gem

- * 产生会员系统的必要文件
- `rails g devise:install`, `rails g devise user`

5-3 让这个网站有实际“登录”、“退出”的功能

5-4 让“群组”与“使用者”产生关联

- * 新增 user_id 到 group 的 table 里(增加model表格栏位的方法)
- * 连结 user 与 group 的双向关系 (一对多)
- * 在新增看板时, 记录谁是群组的建立者(判断, 加入)

5-5 只有群组的“创始者”可以“编辑”“删除”群组资讯

- * 路人不应该可以看到“编辑”“删除”按钮
- * 路人不应该也可以“直接输入网址”去存取 edit / update / destroy action
- * 制作 find_group_and_check_permission
- * 把 find_group_and_check_permission 挂到 before_action
- * 修掉 show 里面的 Edit 按钮

6-2 建立文章

- * 设计 Post 的 model 架构
- * 建立 Group / Post / User 三者间的关系
- * 建立 Post Controller
- * 设立 routing
- * 在 groups#show 新增 “发表文章”按钮
- * 实作 new / create action
- * 修改 groups_controller 中的 show action, 让文章能够显示出来
- * 限制 post 的 content 不能为空

6-4 文章应该按照发表时间倒序排列

- * 修改 groups_controller
- * 使用 scope 替代直接 order

6-5 加入文章分页功能

- * 安装 will_paginate ([Github: Will_paginate](https://github.com/mislav/will_paginate/wiki))

7. 7-2 建立“群成员”数据表

- * 建立 GroupRelationship model
- * 设定 Group 与 User 之间的关系 (使用者参与的所有群)
- * 设定 Group 与 User 之间的关系 (群组内的所有会员)
- * 在 user model 内实作判断式“是否为群组的一分子”
- * 一个使用者可以选择“加入”、“退出”讨论群
- * 群的创始者, 创群一开始就应该加在群组里

7-5 实际操作“加入群组”或“退出群组”

- * 加入与退出群组必须要是登入状态下才行
- * app/controllers/groups_controller.rb

- * config/routes.rb
- * app/views/groups/show.html.erb

8-2 可以看到自己参与的所有群组

- * 产生 account 的 namespace 下的 groups_controller
- * 修改 routing
- * 修改下拉选项
- * 建立 account/groups_controller.rb 下的 index action
- * 新增 “参与群组一览表”

8-3 可以看到自己发表的所有文章

- * 产生 account 的 namespace 下的 posts_controller
- * 建立 account/posts_controller.rb 下的 index action
- * 修改 routing
- * 新增 “发表文章一览表”

9-2 系统内建 Helper “simple_format”的使用

- * 将 group.description 改成 simple_format(group.description) (simple_format 是 Rails 内建的 Helper, 可以帮忙把 \r\n 翻译成为 \
)

9-3 使用自定义 Helper

- * 自制 render_group_description(group)

9-4 partial 还可以这样用

- * 修改 app/views/groups/index.html.erb
- * <%= render :partial => "group_item", :collection => @groups, :as => :group %>