



第4章

背景案例

本章内容

- ☐ 简单的账户注册服务
- ☐ 需求阐述
- ☐ 简要设计
- ☐ 小结



前几章已经大概解释了 Maven 是什么，并且介绍了 Maven 的安装和最基本的使用。从本章开始，引入一个较为真实的背景案例，以演示 Maven 使用的真实场景。由于本书的主题是 Maven，我们不想让项目变得过于复杂，或者涉及过多的技术，因此该案例的目的还是帮助我们理解 Maven 的概念，以及展示大部分 Maven 项目需要面对和处理的一些问题。

建议读者至少大概浏览本章内容，因为本章是几乎所有后续章节的背景，了解了背景需求，将能够更好地理解相关 Maven 概念及实践的阐述。

4.1 简单的账户注册服务

注册互联网账户是日常生活中再熟悉不过的一件事情，作为一个用户，注册账户的时候往往需要做以下事情：

- ☐ 提供一个未被使用的账号 ID
- ☐ 提供一个未被使用的 email 地址
- ☐ 提供一个任意的显示名称
- ☐ 设置安全密码，并重复输入以确认
- ☐ 输入验证码
- ☐ 前往邮箱查收激活链接并单击激活账号
- ☐ 登录

账号的 ID 和 email 地址都可以用来唯一地标识某个账户，而显示名称则用来显示在页面上，方便浏览。注册的时候用户还需要输入两次密码，以确保没有输错。系统则需要负责检查 ID 和 email 的唯一性，验证两次输入的密码是否一致。验证码是由系统随机生成的只能由肉眼识别其内容的图片，可以有效防止机器恶意批量注册，若输入正确的验证码信息，系统则会进行检查，如果验证码错误，系统会生成并返回新的验证码。一旦所有检查都没问题了，系统就会生成一个激活链接，并发送到用户的邮箱中。单击激活链接后，账户就被激活了，这时账户注册完成，用户可以进行登录。

对于一个账户注册服务，还需要考虑一些安全因素。例如，需要在服务器端密文地保存密码，检查密码的强弱程度，更进一步则需要考虑验证码的失效时间，激活链接的失效时间，等等。

本章的主要目的是让读者清楚地了解这个背景案例，即账户注册服务，它的需求是什么，基于这样的一个需求，我们会怎样设计这个小型的系统。本章的描述几乎不会涉及 Maven，但后面的章节在讲述各种 Maven 概念和实践的时候，都会基于这一实际的背景案例。

4.2 需求阐述

了解账户注册服务之后，下面从软件工程的视角来分析一下该服务的需求。

4.2.1 需求用例

为了帮助读者详细地了解账户注册服务的需求，这里正式阐述一下账户注册服务的需求用例，见图 4-1。

注册账户

主要场景:

1. 用户访问注册页面
2. 系统生成验证码图片
3. 用户输入想要的 ID、Email 地址, 想要的显示名称、密码、确认密码
4. 用户输入验证码
5. 用户提交注册请求
6. 系统检查验证码
7. 系统检查 ID 是否已经被注册, Email 是否已经被注册, 密码和确认密码是否一致
8. 系统保存未激活的账户信息
9. 系统生成激活链接, 并发送至用户邮箱
10. 用户打开邮箱, 访问激活链接
11. 系统解析激活链接, 激活相关账户
12. 用户使用 ID 和密码登录

扩展场景:

- 4a: 用户无法看清验证码, 请求重新生成
 1. 跳转到步骤 2
- 6a: 系统检测到用户输入的验证码错误
 1. 系统提示验证码错误
 2. 跳转到步骤 2
- 7a: 系统检测到 ID 已被注册, 或者 Email 已被注册, 或者密码和确认密码不一致
 1. 系统提示相关错误信息
 2. 跳转到步骤 2

图 4-1 账户注册服务需求用例

该注册账户用例包含了一个主要场景和几个扩展场景。该用例的角色只有两个：用户和系统。“主要场景”描述了用户如何与系统一步一步地交互，并且成功完成注册。“扩展场景”则描述了一些中途发生意外的情形，比如用户输错验证码的时候，系统就需要重新生成验证码，用户也需要重新输入验证码。

该用例没有涉及非功能性需求（如安全性），也没有详细定义用户界面，用例也不会告诉我们使用何种技术。关于该服务的安全性，你将会看到一些实际的措施，但我们不会过于深入；关于用户界面，下一小节会给出一个界面原型；至于使用的技术，该项目会基于大家所熟知的 Spring 进一步开发。

4.2.2 界面原型

虽然根据图 4-1 中的文字描述，我们已经了解了用户注册服务所涉及的内容，但图 4-2 所示的注册页面更加直观。图 4-2 清楚地标示了注册账户所需要填写的各个字段，还展示了一个验证码图片，旁边还有一个简单的链接用来获取新的验证码图片。

注册账户

http://localhost:8080/service/signup

注册新账户

账户 ID: juven ?

Email: juven@changeme.com ?

显示名称: Juven Xu ?

密码: ***** ?

确认密码: ***** ?

验证码: nn5dde ?

看不清? 换一张

确认并提交

图 4-2 注册账户服务界面原型

4.3 简要设计

4.3.1 接口

详细了解了这个简单账户注册服务的需求之后，就能勾勒出该系统对外的接口。从需求用例中可以看到，系统对外的接口包括生成验证码图片、处理注册请求、激活账户以及处理登录等。图4-3描述了系统的接口。

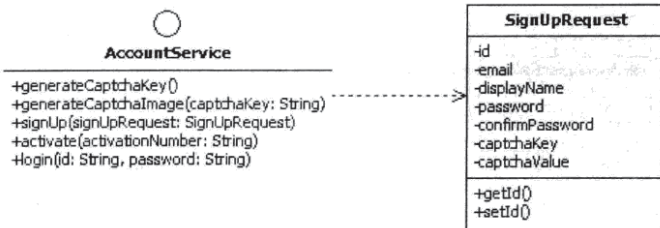


图4-3 注册账户服务系统接口

首先需要解释的是 `generateCaptchaKey()` 和 `generateCaptchaImage()` 方法，对于 Captcha 的简单解释就是验证码。每个 Captcha 都需要有一个 key，根据这个 key，系统才能得到对应的验证码图片以及实际值。因此，`generateCaptchaKey()` 会生成一个 Captcha key，使用这个 key 再调用 `generateImage()` 方法就能得到验证码图片。验证码的 key 以及验证码图片被传送到客户端，用户通过肉眼识别再输入验证码的值，伴随着 key 再传送到服务器端验证，服务器端就可以通过这个 key 查到正确的验证码值，并与客户端传过来的值进行比对验证。

`SignUpRequest` 包含了注册用户所需要的信息，包括 ID、email、显示名称、密码、确认密码等。这些信息伴随着 Captcha key 和 Captcha value 构成了一个注册请求，`signUp()` 方法接收 `SignUpRequest` 对象，进行验证，如果验证正确，则创建一个未被激活的账户，同时在后台也需要发送一封带有激活链接的邮件。

`activate()` 方法接收一个激活码，查找对应的账户进行激活。

账户激活之后，用户可以使用 `login()` 方法进行登录。

4.3.2 模块结构

定义了系统核心的接口之后，基于功能分割和方便复用的原则，再对系统进一步进行划分。这里基于包名划分模块，这也是在 Java 中比较常见的做法。

也许你会觉得为如此简单的一个系统（或许根本就不该称之为系统）划分模块有点小题大做了，有经验的程序员根本不需要多少设计就能快速完成这样的一个注册功能。不过

本书的目的不在这个功能本身，我们需要一个像模像样的、有很多模块的系统来演示 Maven 很多非常酷的特性，同时，又不想引入一个拥有成千上万行代码的过于庞大的系统。账户注册服务的模块划分如图 4-4 所示。

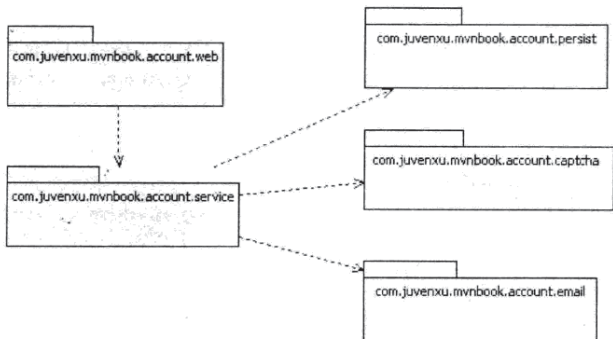


图 4-4 注册账户服务包图

现在逐个解释一下各个模块（包）的作用：

- ❑ `com.juvenxu.mvnbook.account.service`：系统的核心，它封装了所有下层细节，对外暴露简单的接口。这实际上是一个 Facade 模式，了解设计模式的读者应该能马上理解。
- ❑ `com.juvenxu.mvnbook.account.web`：顾名思义，该模块包含所有与 web 相关的内容，包括可能的 JSP、Servlet、web.xml 等，它直接依赖于 `com.juvenxu.mvnbook.account.service` 模块，使用其提供的服务。
- ❑ `com.juvenxu.mvnbook.account.persist`：处理账户信息的持久化，包括增、删、改、查等，根据实现，可以基于数据库或者文件。
- ❑ `com.juvenxu.mvnbook.account.captcha`：处理验证码的 key 生成、图片生成以及验证等，这里需要第三方的类库来帮助实现这些功能。
- ❑ `com.juvenxu.mvnbook.account.email`：处理邮件服务的配置、激活邮件的编写和发送等工作。

4.4 小结

到目前为止，我们已经了解了账户注册服务的需求、大概的界面、简单的接口设计以及模块的职责划分，虽然我们没有实际编写代码，但这已足够支持本书后续章节关于 Maven 概念和实践的描述。在下面的章节中，这个简单的账户注册服务将得以一步步地实现和完善，同时我们也将看到 Maven 如何与实际项目结合并发挥自己的功效。