

48_谋定后动：先画图分析如何基于 Spring Event 对登录系统 核心流程进行异步化改造

1、开篇

上节课主要介绍了小程序的登录验证，其中包括：登录小程序、获取手机号和解密手机号三个部分。本节课还将继续登录验证的内容，会基于核心登录流程将中间的步骤进行异步化的操作，特别是对登录的信息保存和发放优惠券的动作进行解耦。

今天的内容包括以下几个部分：

- 传统登录流程
- 异步化登录流程

2、传统登录流程

如果要先谈异步处理流程就要先看看传统登录流程是怎样的，由于该系统有两种发放优惠券的模式：首次登陆优惠券和生日优惠券，这里我们以首次登陆优惠券为例给大家讲解。如图 1 所示，我们把登录流程进行简化，并且把焦点放到“小程序后台”上。当用户登录的时候，会判断“是否第一次登录？”当用户满足第一次登录条件以后，后台会在数据库中记录登录的信息，然后发放优惠券。

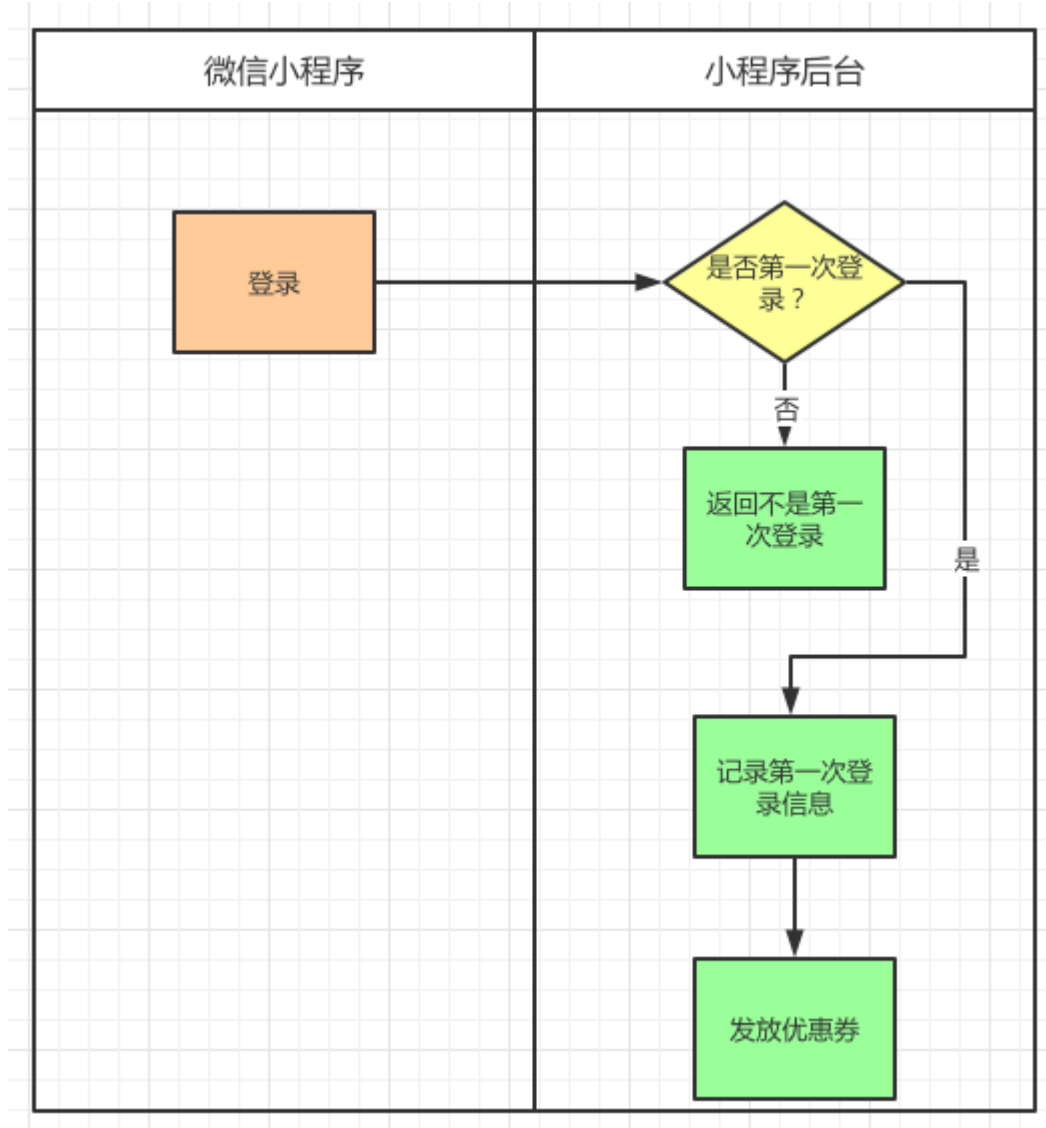


图 1 传统登录流程

需要注意的是，“记录第一次登录信息”和“发放优惠券”在执行上是有先后顺序的，为了方便也会把这两个功能放到一个模块里面执行。

但为了服务的高内聚、低耦合，或者通过分布式提高服务高并发能力，会将模块或者服务器切分，例如：将登录模块和优惠券模块拆分成两个服务。服务拆分以后，服务内部保证实现对应的功能，服务的执行先后顺序可以根据业务需求进行调整。例如：可以在“记录第一次登录信息”的同时就“发放优惠券”，不用等待记录成功也可以进行优惠券的发放。让系统效率最优，同时让更多的服务之间可以进行组合。

3、异步化登录流程

上面介绍了传统登录的流程，大家会发现是通过同步方式进行的，登录完毕更新状态再发送优惠券，同步的方式本身就带来性能问题。为了提升登录接口的性能，同时兼顾服务的高内聚、低耦合，分布式的要求，就有了异步化的登录流程。

如图 2 所示，我们将登陆的整个过程通过 **ConsumerService**、**LogonEventPublisher**、**Spring Event** 以及 **LogonListener** 串联起来。从左往右来看这张图，首先学员通过微信小程序“登陆”系统，通过 **ConsumerService** 提供的登陆功能，该功能中会判断是否第一次登陆，并且记录用户的登陆信息。随后，通过 **LogonEventPublisher** 发布用户的登陆的消息，该消息通过 **Spring Event** 的消息对立让 **LogonListener** 的“登陆消息消费者”接收到，然后再更新用户登陆的次数。整个过程通过 **Spring Event** 消息队列进解耦，将登陆验证和登陆次数的更新变成两个异步操作。

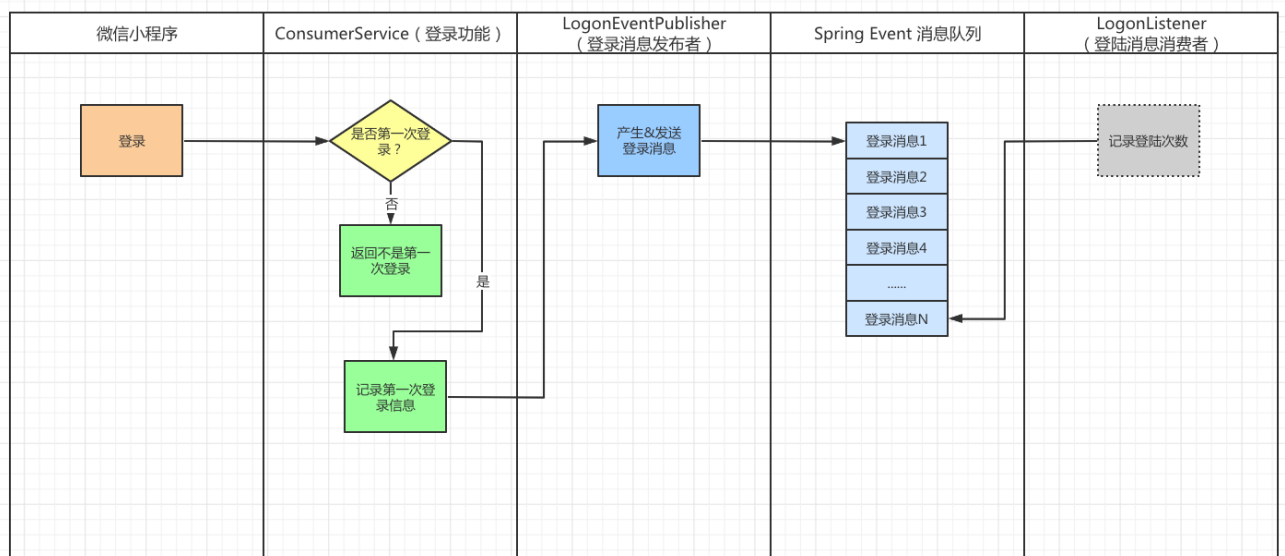


图 2 异步化登录流程

理解了通过 **Spring Event** 对登陆进行异步化的操作以后，后面会根据这个流程进行代码的编写。

4、总结

本节课梳理了登陆系统的异步流程，同时解决了登陆异步消息处理的问题，将登陆和登陆次数的记录通过 **Spring Event** 变成两个异步操作。下节课会在这个流程的基础上进行代码的实现。下期见，拜拜。