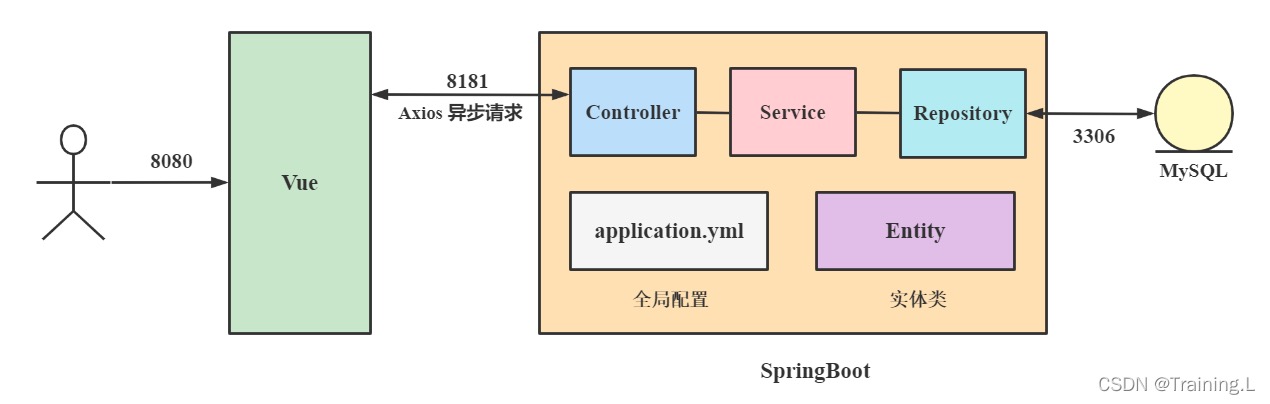
我们的外卖订单系统项目采用的是Spring Boot+Vue前后端分离的两层架构。前端使用vue,后端使用springboot集成mybatis。

项目流程如下：



Sptingboot实现了和spring data JPA的整合实现后端数据交互。Spring Boot 可以尽可能快的跑起来 Spring 应用程序并且尽可能减少的配置文件。

前后端分离：实现真正的前后端解耦。核心思想是前端html页面通过ajax调用后端的restuful api接口并使用json数据进行交互。前后端分离会为以后的大型分布式架构、弹性计算架构、微服务架构、多端化服务（多种客户端，例如：浏览器，安卓，IOS等等）打下坚实的基础。适合业务逻辑复杂的项目或庞大的项目，若项目较小或者修改变动较小，可以考虑前后端不分离，不过现在的硬件空间便宜，比较建议前后端分离。前后端分离的优点有：

1）前后端解耦，各自职责明确，分工合理，开发效率较高。

2）减轻服务器压力，各自配置，各自运作，更容易维护。

3）项目模块化更明显，业务逻辑更加清晰，不至于等到项目规模过大的时候无法维护或者更新项目。

4）方便单元测试以及bug调试，互不相干，前后端的人技术职责分开，减少前后端的人打架的概率，技术成本更少...

但也存在缺点：

1）搭建项目成本相对高一些，因为需要前后端项目同时运作。

2）对于项目的初步设计需要有很好的文档说明，大家都按照一定的约定来开发，这就考验项目技术经理的能力了。

而Vue，是一套用于构建用户界面的渐进式框架，在github十分受欢迎。不需要操作Dom，实现了MVVM，且学习成本低，文档浅显易懂。

Vue的生命周期如下图：

