

平台架构分析

上海成趣信息科技有限公司 杨晓春

- 什么是平台 Platform?
- 什么是框架 Framework?
- 什么是架构 Architecture?
- 选择框架的基本原则
- 分析比较 Spring MVC vs Spring Boot
- 读Spring代码，画UML图
- 如何做平台技术评估?

什么是平台？

- 系统平台（英语：computing platform）是指在计算机里让软件运行的系统环境，包括硬件环境和软件环境。典型的系统平台包括一台计算机的硬件体系结构（computer architecture）、操作系统、运行时库等。
- https://en.wikipedia.org/wiki/Computing_platform

什么是框架？

- 软件框架（software framework），通常指的是为了实现某个业界标准或完成特定基本任务的软件组件规范，也指为了实现某个软件组件规范时，提供规范所要求之基础功能的软件产品。
- 框架的功能类似于基础设施，与具体的软件应用无关，但是提供并实现最为基础的软件架构和体系。软件开发者通常依据特定的框架实现更为复杂的商业运用和业务逻辑。这样的软件应用可以在支持同一种框架的软件系统中运行。
- 简而言之，框架就是制定一套规范或者规则（思想），大家（程序员）在该规范或者规则（思想）下工作。或者说使用别人搭好的舞台来做编剧和表演。
- https://en.wikipedia.org/wiki/Software_framework

什么是架构？

- 软件架构是指软件系统的高级结构以及创建这种结构和系统的学科。每个结构都包含软件元素，它们之间的关系，以及元素和关系的属性。软件系统的体系结构是一种隐喻，类似于建筑物的体系结构。它作为系统和开发项目的蓝图，规划了由设计团队执行的必要任务
- https://en.wikipedia.org/wiki/Software_architecture

选择框架的基本原则

- 时机
- 技术特性和功能
- 社区参与
- 选择的后果
- 参考 What are the main features I should consider when choosing a framework?
- Ten criteria for choosing the correct framework

分析比较Spring MVC和Spring Boot

- 参考 <http://www.springboottutorial.com/spring-boot-vs-spring-mvc-vs-spring>
- Spring Framework想要解决的问题是什么？
- 概述Spring MVC Framework
- Spring MVC Framework想要解决的问题是什么？
- 概述Spring Boot Framework
- Spring Boot想要解决的问题是什么？
- 将Spring与Spring Boot与Spring MVC进行比较

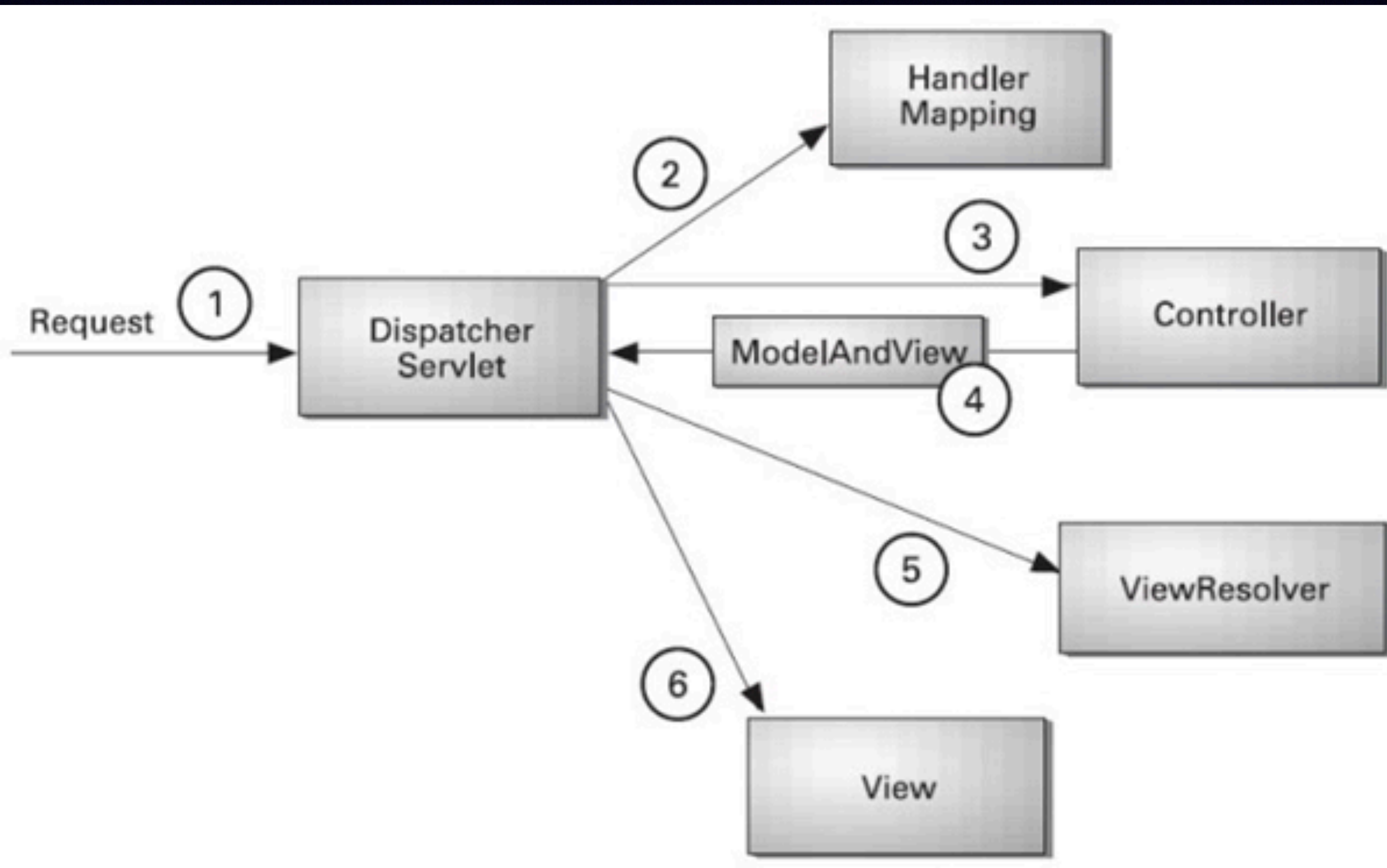
Spring Framework

- Dependency Injection 依赖注入
- IOC Inversion of Control
- 当正确使用DI或IOC时，我们可以开发松散耦合的应用程序
- 松散耦合的应用程序可以轻松进行单元测试
- 复制管道代码
- 与其他框架良好集成

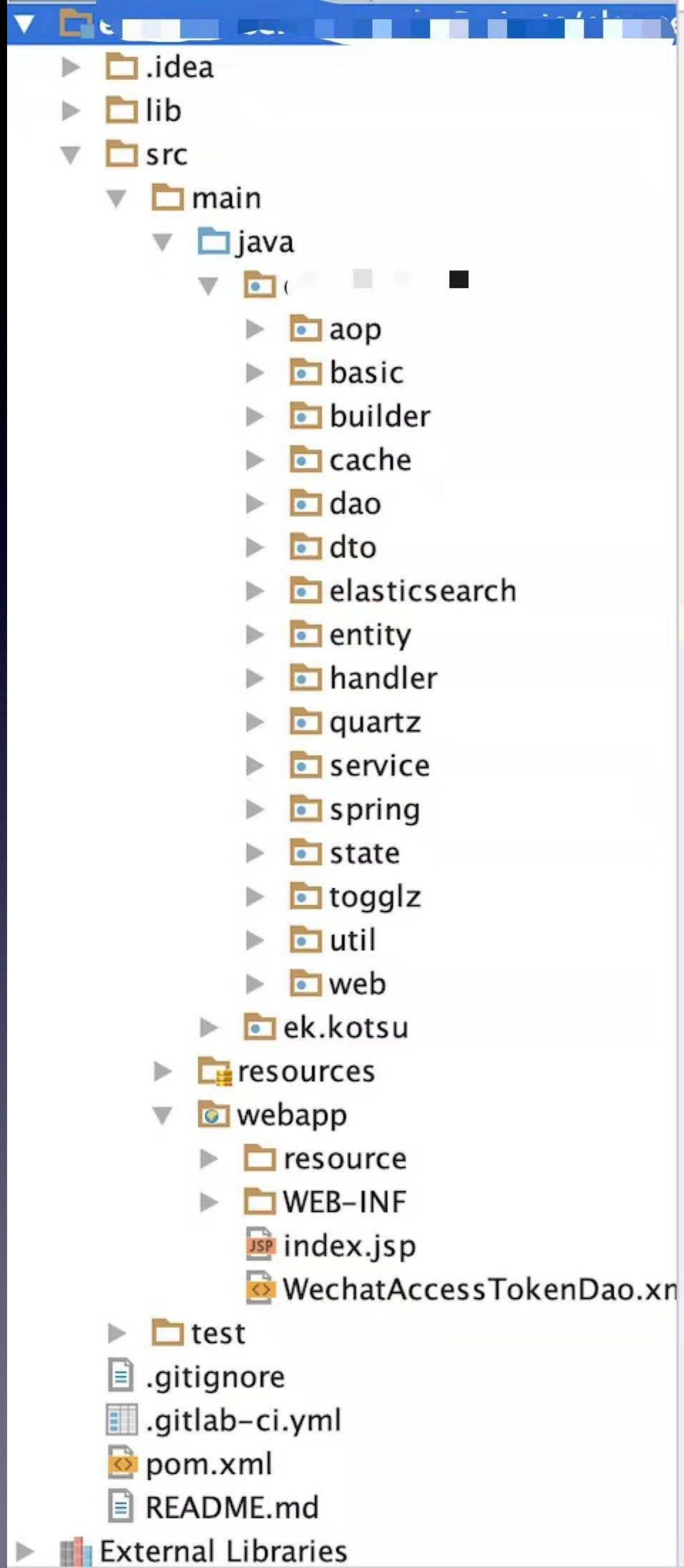
Spring MVC

- Spring MVC Framework提供了开发Web应用程序的分离方式。使用 Dispatcher Servlet, ModelAndView和View Resolver等简单概念, 可以轻松开发Web应用程序。

了解Spring Web MVC的流程



- 所有传入请求都被作为前端控制器的DispatcherServlet拦截。
- DispatcherServlet从XML文件获取处理程序映射条目，并将请求转发给控制器。
- 控制器返回ModelAndView的对象。
- DispatcherServlet检查XML文件中视图解析器的条目并调用指定的视图组件。



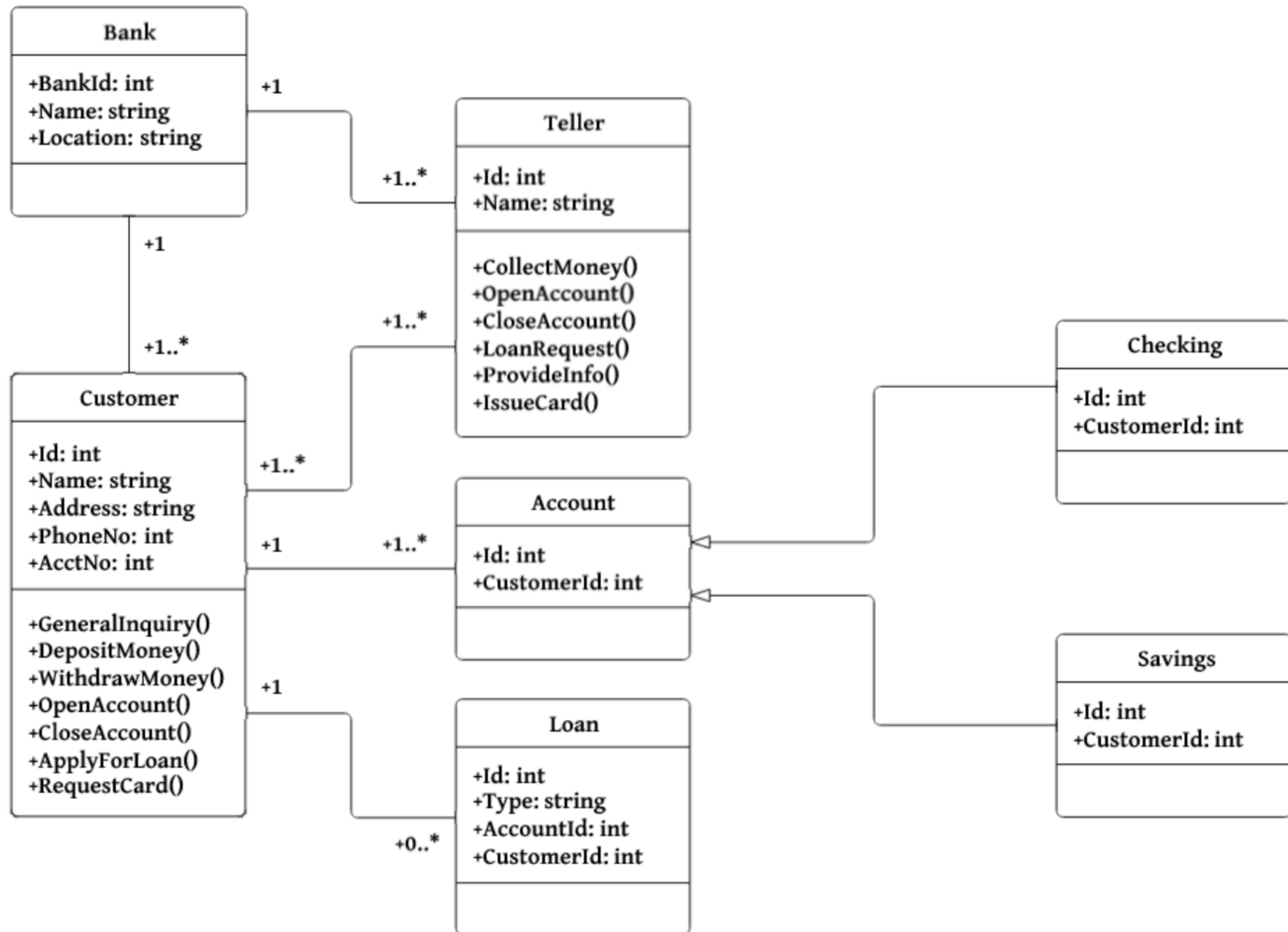
- 一个用Spring MVC框架的代码结构

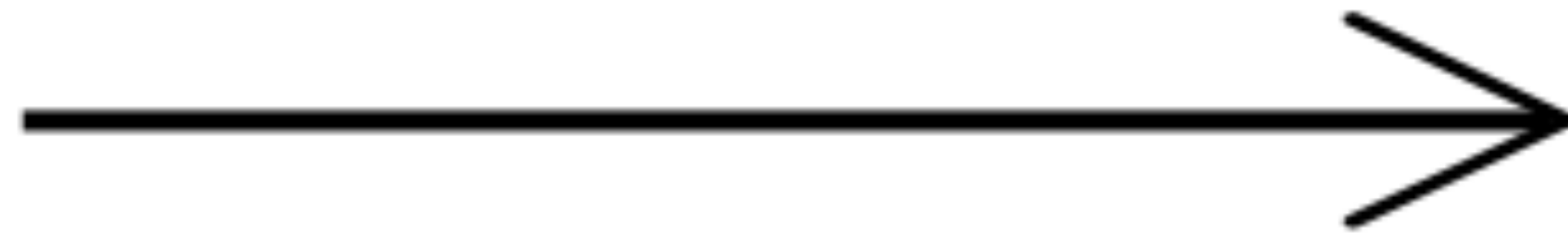
Spring Boot

- 自动配置：Spring Boot查看a) CLASSPATH上可用的框架b) 应用程序的现有配置。基于这些，Spring Boot提供了使用这些框架配置应用程序所需的基本配置
- Spring Boot Starter项目：围绕众所周知的模式构建。启动器是一组方便的依赖关系描述符，可以在应用程序中包含这些描述符，获得所需的所有Spring和相关技术的一站式服务，而无需搜索示例代码并复制粘贴的依赖描述符。例如：`spring-boot-starter-data-jpa`
- 旨在快速实现产品环境就绪（Production ready）应用程序

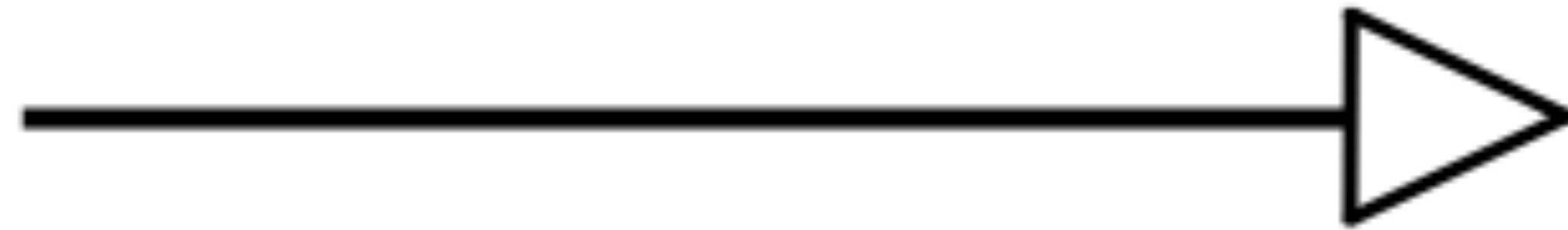
为什么使用UML?

- 用例图
- 类图 and 对象图
- 交互图
 - 顺序图
 - 通信图
 - 定时图
- 状态图
- 活动图
 - 基本活动图
 - 带泳道的活动图
 - 交互概览图
- 构件图
- 部署图

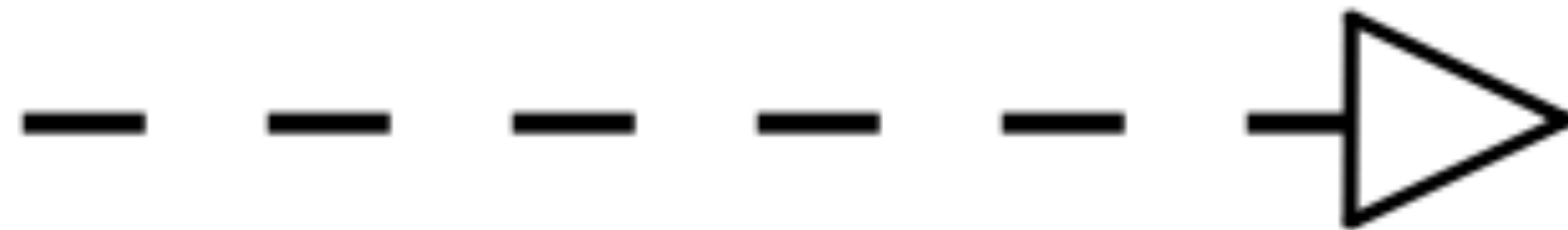




Association



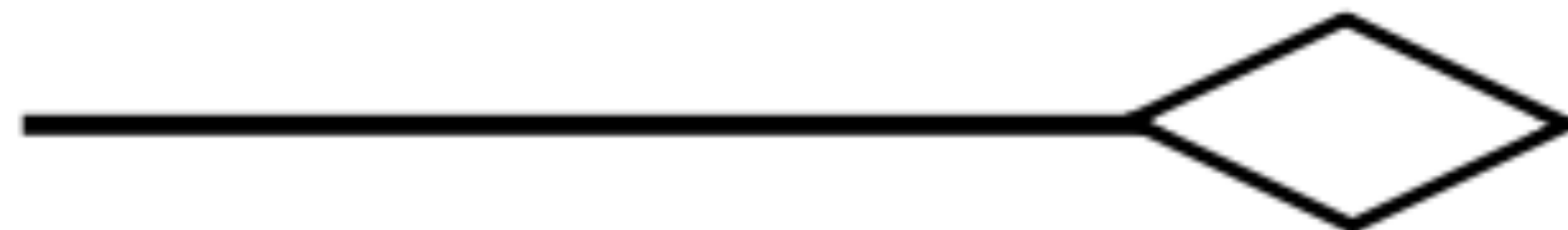
Inheritance



Realization /
Implementation



Dependency



Aggregation



Composition

为什么画类图？

- 提前规划和建模使编程更容易。
- 对类图进行更改很容易，而基于事实编写不同的功能则很烦人。
- 类图是蓝图和设计计划，便于分析和修改系统。
- 不需要太多技术/语言特定的知识来理解它。

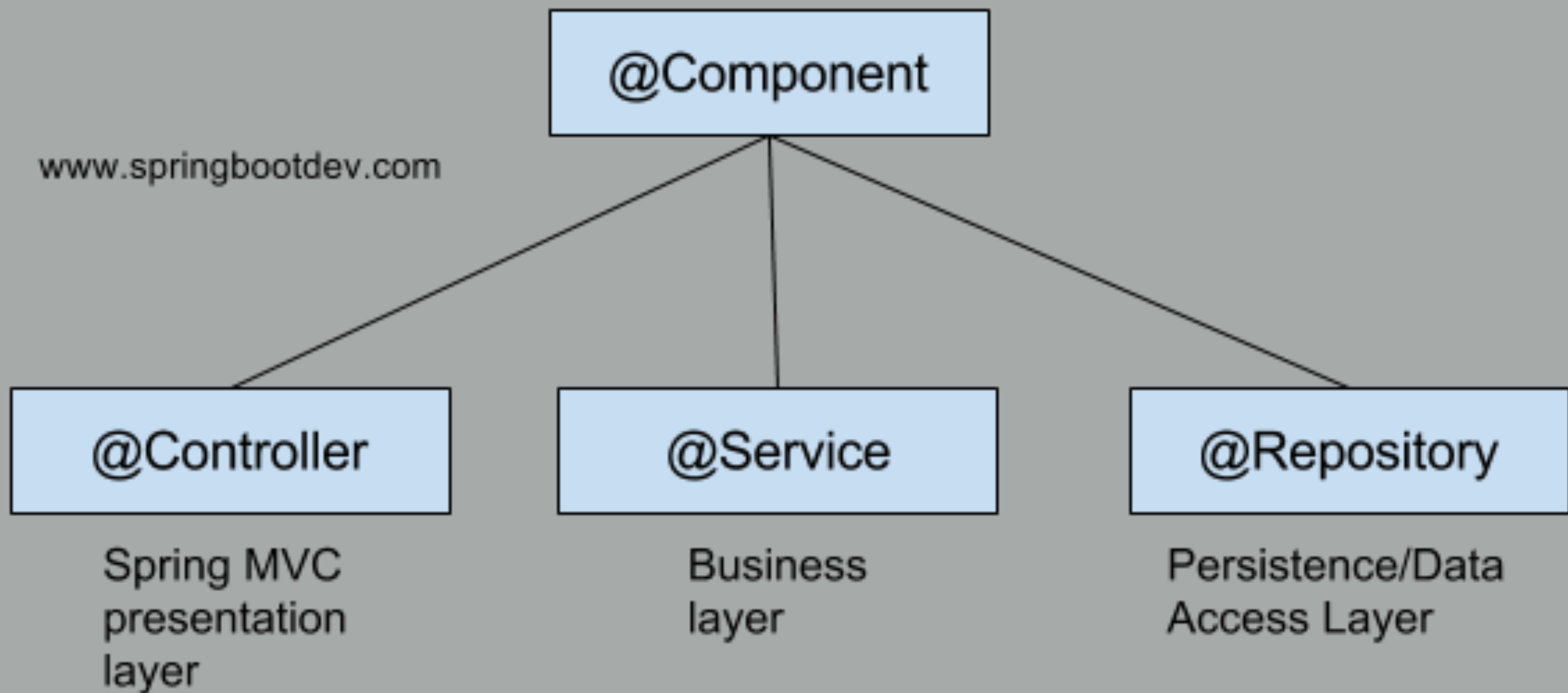
如何画UML图？

- 阅读代码
- 使用IDE的工具

画哪些图？

- 用例图
- 类图和对象图
- 交互图
- 状态图
- 活动图
- 构件图
- 部署图

www.springbootdev.com



平台技术评估指标（一）

- 稳定性 Stability
- 可用性 Availability
- 安全性 Security
- 扩展性 Scalability
- 技术成熟 Maturity
- 效能 Performance

平台技术评估指标（二）

- 相互操控能力 Inter-operability
- 功能 Functionality
- 技术支持 Technical Support
- 技术获得的普遍性 Skills Availability
- 技术发展的远景 Vision
- 技术上的管理 Manageability
- 整体运行成本 Total Cost of Ownership
- 开放性 Openness

课后练习

小报告：如何选择框架？

格式：Word文档，字数不限。

内容：针对报告内容，结合自己要做的
项目，描述选择系统框架的原因和
方法；

发到：janeyungxc@hotmail.com

