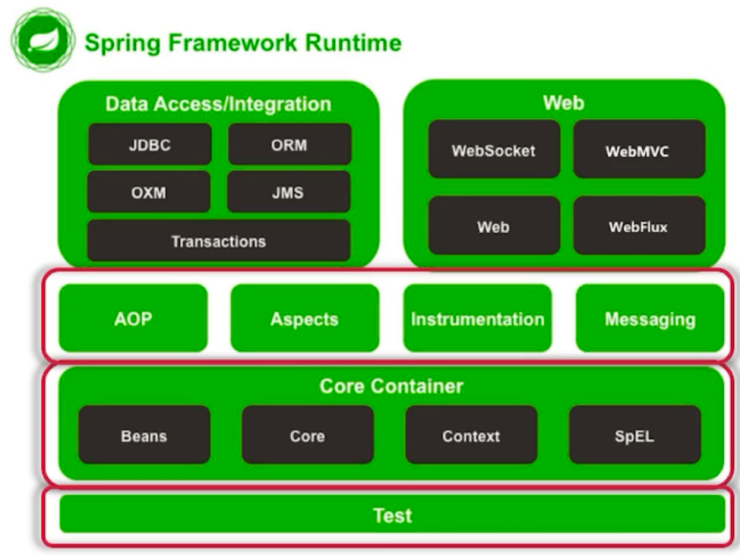
## Spring源码分析

### Spring源码的编译

1. Spring架构



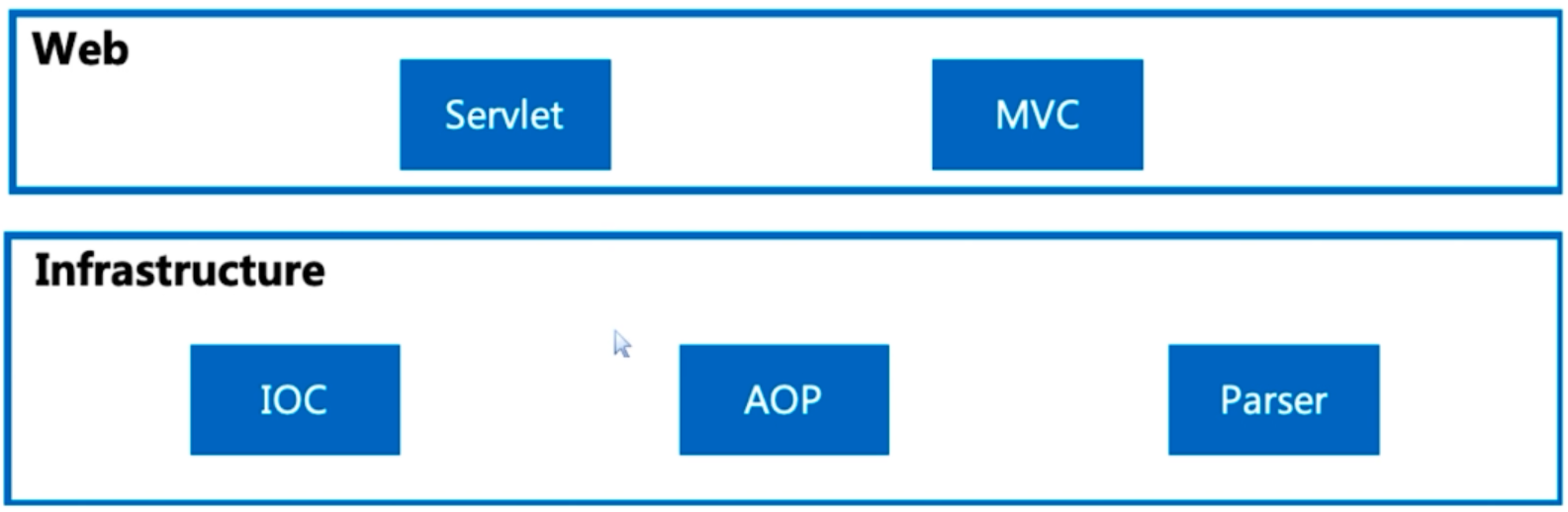
1. Spring核心模块
   1. Spring-core
      1. 包含框架基本的核心工具类，其他组件都会使用到这个包里的类
      2. 定义并提供资源的访问方式
      3. 为IOC & DI 提供最基础的服务
   2. Spring-beans
      1. Spring主要面向Bean编程(BOP)
      2. Bean的定义
      3. Bean的解析
      4. Bean的创建
      5. 重点关注接口——**BeanFactory**
   3. Spring-context
      1. 为Spring提供运行时环境，保持对象的状态
      2. 扩展了BeanFactory
      3. 是Bean之间关系的集合
      4. 重点关注接口——**ApplicationContext**
   4. Spring-aop
      1. 最小化的动态代理实现
      2. JDK动态代理
      3. Cglib
      4. 只能使用运行时织入，仅支持方法级编织，仅支持方法执行切入点
2. 非核心模块
   1. Spring-aspectj + Spring-instrument : Full AspectJ
   2. 织入方式(三种)



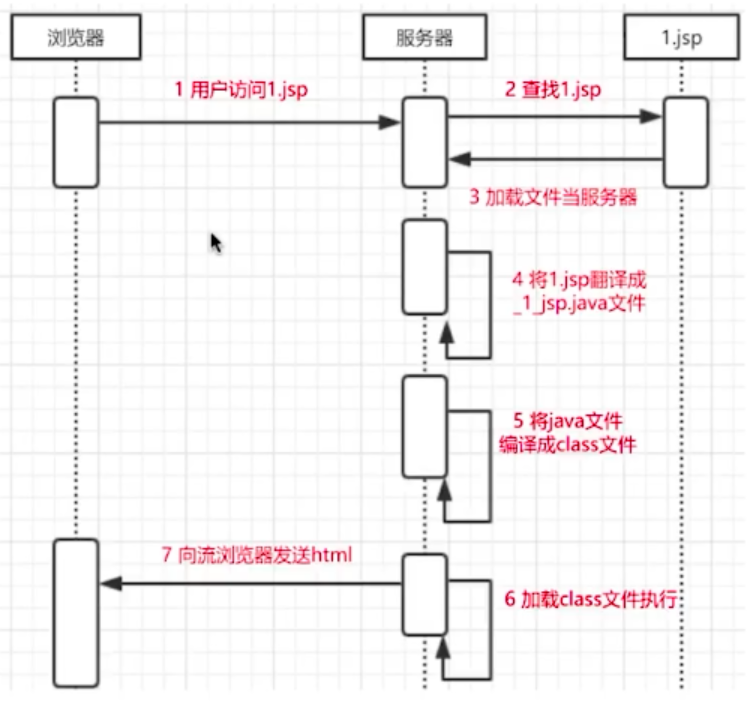
1. 源码编译
   1. [源码下载](https://github.com/spring-projects/spring-framework/tree/v5.2.0.RELEASE)
   2. [IDEA编译步骤](https://github.com/spring-projects/spring-framework/blob/master/import-into-idea.md)
   3. 多阅读[spring-framework](https://docs.spring.io/spring/docs/current/spring-framework-reference/)
      1. 尽量把里面的 Overview 和 Core 两个模块都读一遍
   4. 学习掌握设计模式，熟悉Spring框架的标签和注解的作用
   5. 版本：GA（General Availability）,官方正式发布的稳定版本

### 业务系统架子的构建【自研框架的起源】

1. 自研框架架构简图



1. 创建一个java-web项目
   1. 利用传统的servlet + jsp来处理
   2. JSP的运行原理
      1. 在Tomcat目录下找到work目录，然后跟下去找到[例如：hello\_jsp.java]
      2. 在了解源码文件的前提需要先知道JSP运行原理



### 自研框架IOC实现前奏【从项目开发到框架开发的转换】

### 自研框架IOC容器的实现【实战了解SpringIOC的脉络】

### SpringIOC容器的源码解析

### 详解SpringIOC容器的初始化【refresh方法】

### 精讲SpringIOC容器的依赖注入【攻坚Bean实例】

### 自研框架AOP的讲解和实现【实战了解SpringAOP的脉络】

### SpringAOP的源码解析

### 自研框架MVC的实现【实战了解SpringMVC的脉络】

### SpringMVC流程分析