第一季：SpringBoot2核心技术-基础入门

学习要求

- 熟悉Spring基础
- 熟悉Maven使用

环境要求

- Java8及以上
- Maven 3.3及以上：https://docs.spring.io/spring-boot/docs/current/reference/html/getting-started.html#getting-started-system-requirements <https://docs.spring.io/spring-boot/docs/current/reference/html/getting-started.html#getting-started-system-requirements>

学习资料

- 文档地址： https://www.yuque.com/atguigu/springboot <https://www.yuque.com/atguigu/springboot>
  - 文档不支持旧版本IE、Edge浏览器，请使用chrome或者firefox
- 视频地址： http://www.gulixueyuan.com/ <http://www.gulixueyuan.com/> https://www.bilibili.com/video/BV19K4y1L7MT?p=1 <https://www.bilibili.com/video/BV19K4y1L7MT?p=1>
- 源码地址：https://gitee.com/leifengyang/springboot2 <https://gitee.com/leifengyang/springboot2>

关注作者和知识库后续更新

尚硅谷

程序员标配，人手一...

已关注

SpringBoot2核心...

基于SpringBoot2.3与2....

关注

推荐阅读

05、Web开发

1、SpringMVC自动配置概览Spring Boot provides a...

03、了解自动配置原理

1、SpringBoot特点1.1、依赖管理父项目做依赖管理...

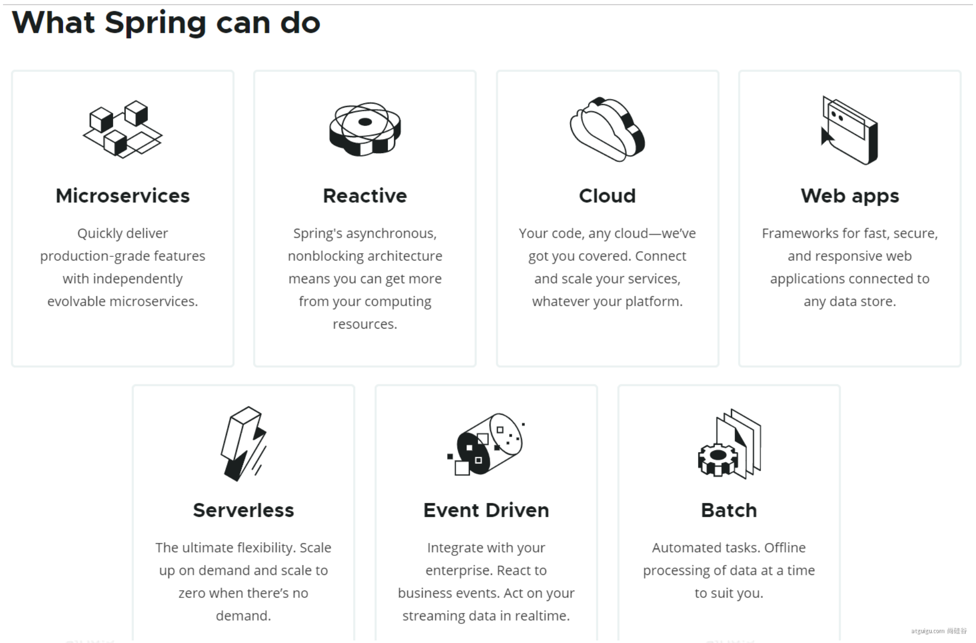
01、Spring与SpringBoot

1、Spring能做什么1.1、Spring的能力1.2、Spring的...

01、Spring与SpringBoot

1、Spring能做什么

1.1、Spring的能力



What Spring can do

Microservices
Quickly deliver production-grade features with independently evolvable microservices.

Reactive
Spring's asynchronous, nonblocking architecture means you can get more from your computing resources.

Cloud
Your code, any cloud—we've got you covered. Connect and scale your services, whatever your platform.

Web apps
Frameworks for fast, secure, and responsive web applications connected to any data store.

Serverless
The ultimate flexibility. Scale up on demand and scale to zero when there's no demand.

Event Driven
Integrate with your enterprise. React to business events. Act on your streaming data in realtime.

Batch
Automated tasks. Offline processing of data at a time to suit you.

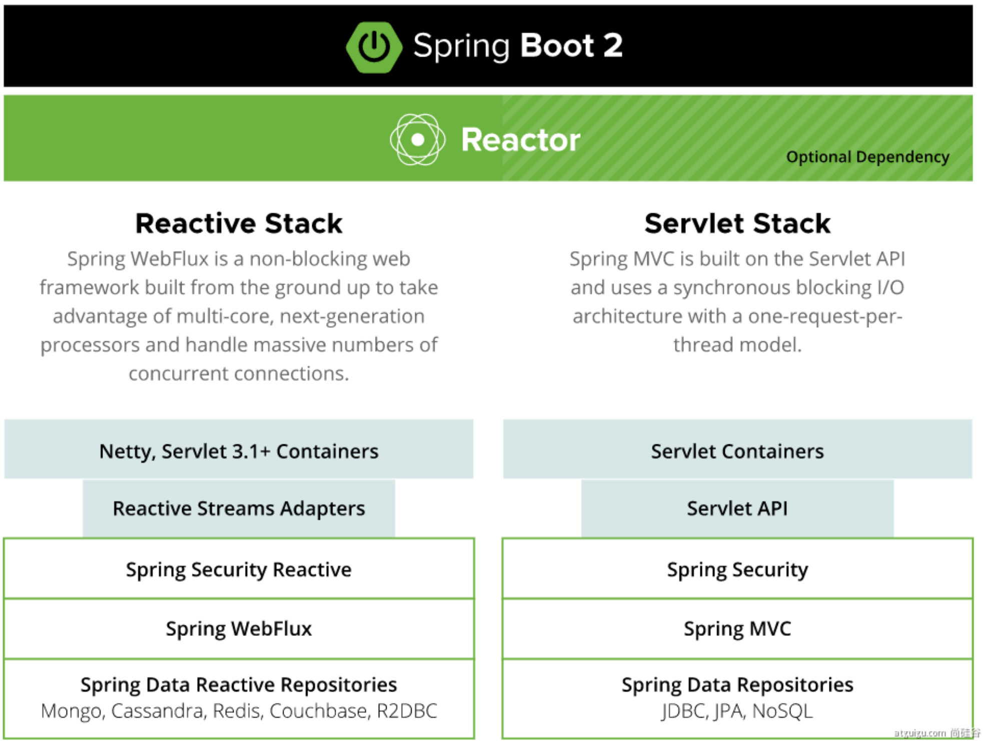
1.2、Spring的生态

https://spring.io/projects/spring-boot <https://spring.io/projects/spring-boot>

覆盖了：
web开发
数据访问
安全控制
分布式
消息服务
移动开发
批处理
......

1.3、Spring5重大升级

1.3.1、响应式编程



Spring WebFlux is a non-blocking web framework built from the ground up to take advantage of multi-core, next-generation processors and handle massive numbers of concurrent connections.

Spring MVC is built on the Servlet API and uses a synchronous blocking I/O architecture with a one-request-per-thread model.

1.3.2、内部源码设计

基于Java8的一些新特性，如：接口默认实现。重新设计源码架构。

2、为什么用SpringBoot

> Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".
>
> 能快速创建出生产级别的Spring应用

2.1、SpringBoot优点

- Create stand-alone Spring applications
  - 创建独立Spring应用
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
  - 内嵌web服务器
- Provide opinionated 'starter' dependencies to simplify your build configuration
  - 自动starter依赖，简化构建配置
- Automatically configure Spring and 3rd party libraries whenever possible
  - 自动配置Spring以及第三方功能
- Provide production-ready features such as metrics, health checks, and externalized configuration
  - 提供生产级别的监控、健康检查及外部化配置
- Absolutely no code generation and no requirement for XML configuration
  - 无代码生成、无需编写XML

> SpringBoot是整合Spring技术栈的一站式框架
> SpringBoot是简化Spring技术栈的快速开发脚手架

2.2、SpringBoot缺点

- 人称版本帝，迭代快，需要时刻关注变化
- 封装太深，内部原理复杂，不容易精通

3、时代背景

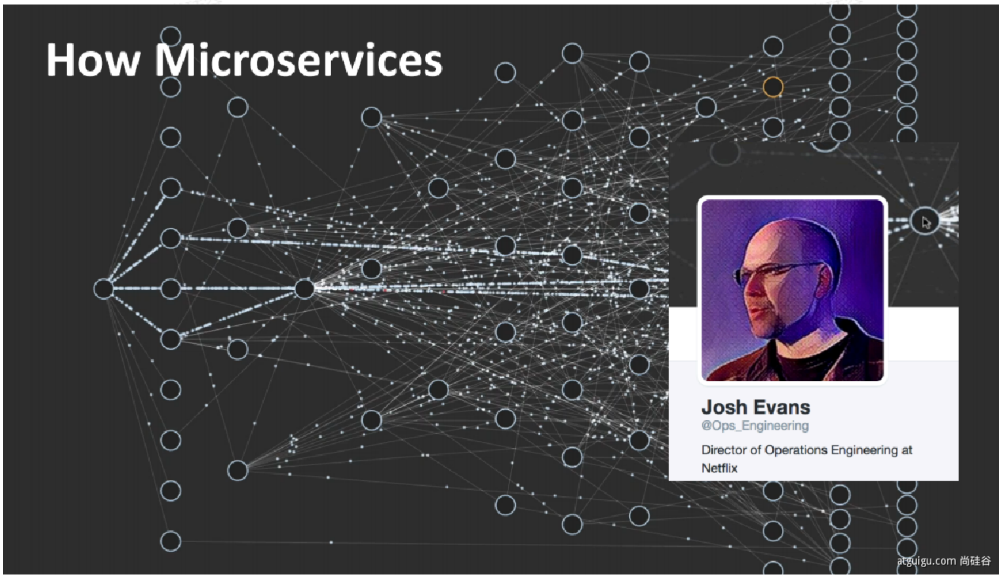
3.1、微服务

James Lewis and Martin Fowler (2014) <https://martinfowler.com/articles/microservices.html> 提出微服务完整概念。https://martinfowler.com/microservices/ <https://martinfowler.com/microservices/>

> In short, the **microservice architectural style** is an approach to developing a single application as a **suite of small services**, each **running in its own process** and communicating with **lightweight** mechanisms, often an **HTTP** resource API. These services are **built around business capabilities** and **independently deployable** by fully **automated deployment** machinery. There is a **bare minimum of centralized management** of these services, which may be **written in different programming languages** and use different data storage technologies.-- James Lewis and Martin Fowler (2014) <https://martinfowler.com/articles/microservices.html>

- 微服务是一种架构风格
- 一个应用拆分为一组小型服务
- 每个服务运行在自己的进程内，也就是可独立部署和升级
- 服务之间使用轻量级HTTP交互
- 服务围绕业务功能拆分
- 可以由全自动部署机制独立部署
- 去中心化，服务自治。服务可以使用不同的语言、不同的存储技术

3.2、分布式

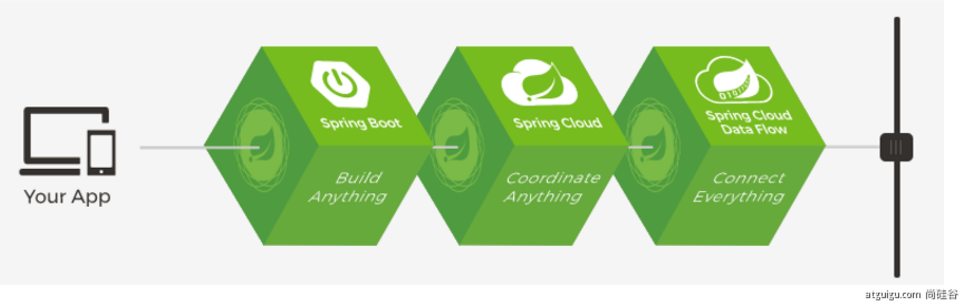


分布式的困难

- 远程调用
- 服务发现
- 负载均衡
- 服务容错
- 配置管理
- 服务监控
- 链路追踪
- 日志管理
- 任务调度
- ......

分布式的解决

- SpringBoot + SpringCloud



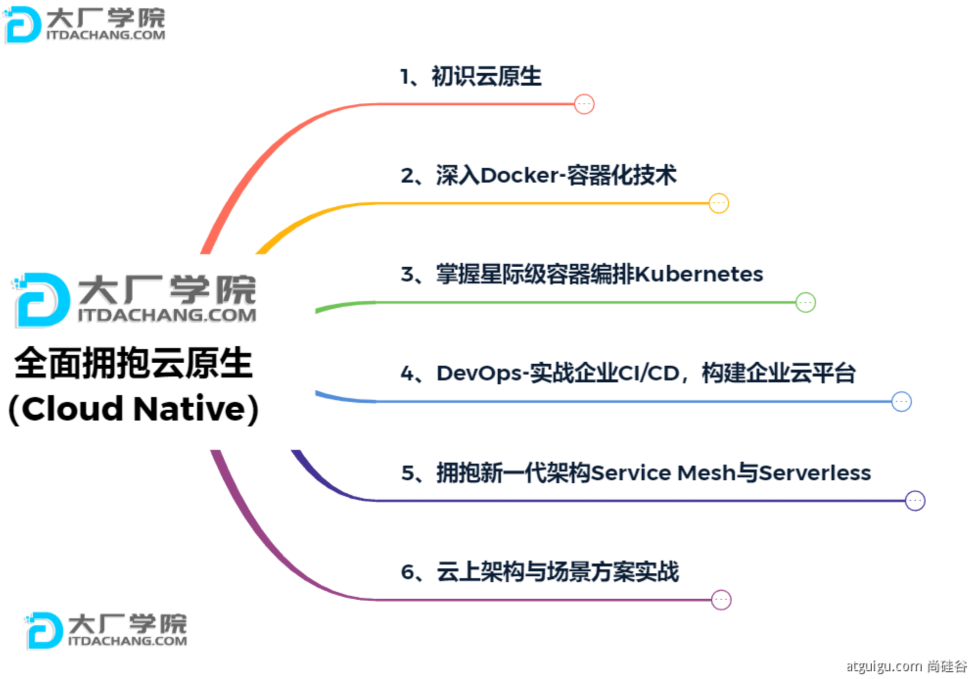
3.3、云原生

原生应用如何上云。 Cloud Native

上云的困难

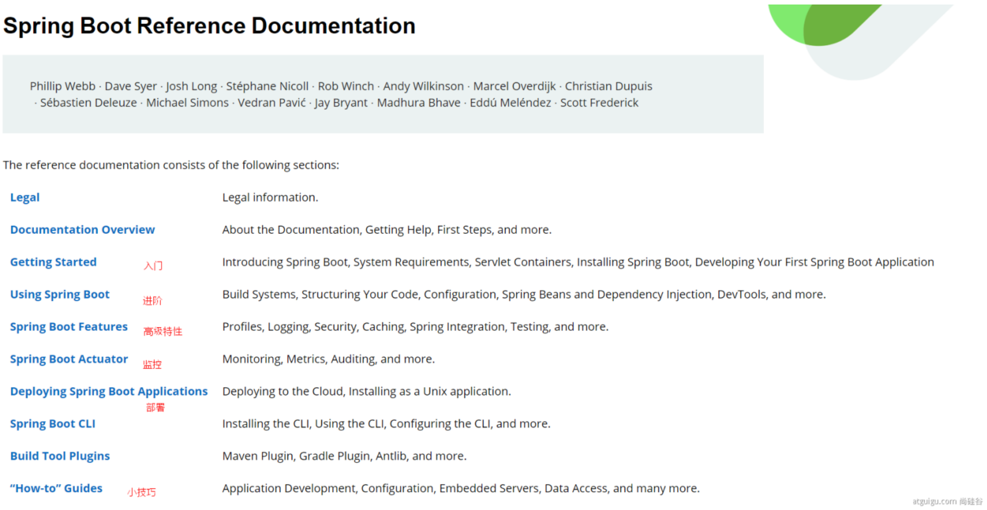
- 服务自愈
- 弹性伸缩
- 服务隔离
- 自动化部署
- 灰度发布
- 流量治理
- ......

上云的解决


大厂学院 ITDACHANG.COM
全面拥抱云原生 (Cloud Native)
1、初识云原生
2、深入Docker-容器化技术
3、掌握星际级容器编排Kubernetes
4、DevOps-实战企业CI/CD，构建企业云平台
5、拥抱新一代架构Service Mesh与Serverless
6、云上架构与场景方案实战
atguigu.com 尚硅谷

4、如何学习SpringBoot

4.1、官网文档架构

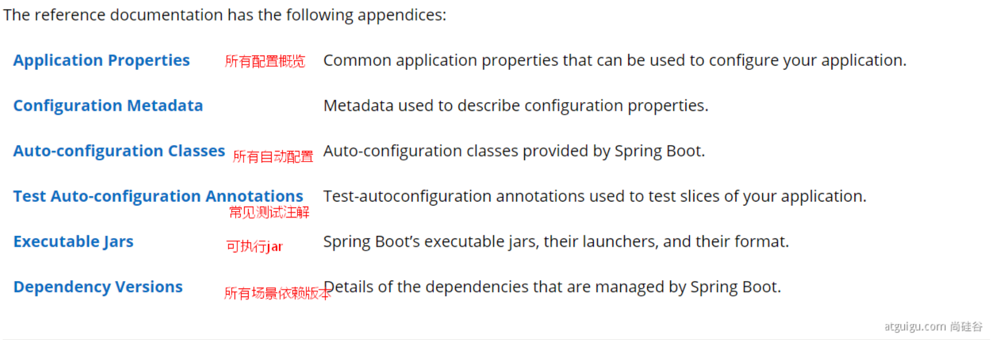


Spring Boot Reference Documentation

Phillip Webb · Dave Syer · Josh Long · Stéphane Nicoll · Rob Winch · Andy Wilkinson · Marcel Overdijk · Christian Dupuis · Sébastien Deleuze · Michael Simons · Vedran Pavić · Jay Bryant · Madhura Bhave · Eddú Meléndez · Scott Frederick

The reference documentation consists of the following sections:

Legal		Legal information.
Documentation Overview		About the Documentation, Getting Help, First Steps, and more.
Getting Started	入门	Introducing Spring Boot, System Requirements, Servlet Containers, Installing Spring Boot, Developing Your First Spring Boot Application
Using Spring Boot	进阶	Build Systems, Structuring Your Code, Configuration, Spring Beans and Dependency Injection, DevTools, and more.
Spring Boot Features	高级特性	Profiles, Logging, Security, Caching, Spring Integration, Testing, and more.
Spring Boot Actuator	监控	Monitoring, Metrics, Auditing, and more.
Deploying Spring Boot Applications	部署	Deploying to the Cloud, Installing as a Unix application.
Spring Boot CLI		Installing the CLI, Using the CLI, Configuring the CLI, and more.
Build Tool Plugins		Maven Plugin, Gradle Plugin, Antlib, and more.
"How-to" Guides	小技巧	Application Development, Configuration, Embedded Servers, Data Access, and many more.

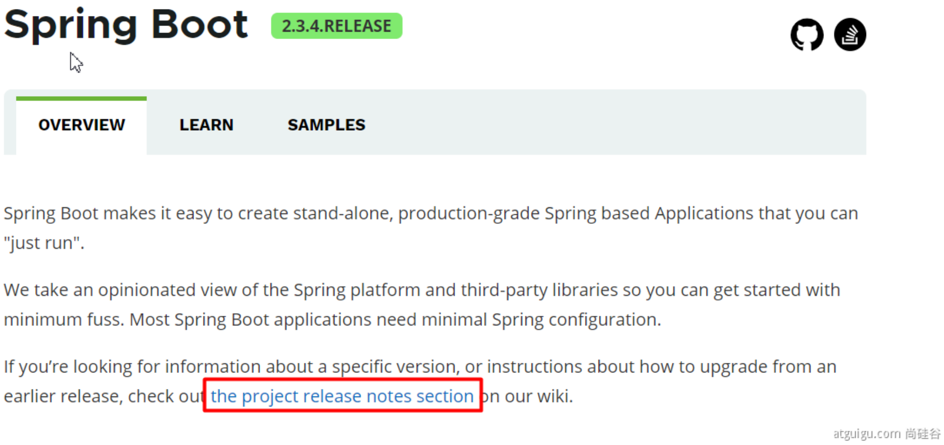


The reference documentation has the following appendices:

Application Properties	所有配置概览	Common application properties that can be used to configure your application.
Configuration Metadata		Metadata used to describe configuration properties.
Auto-configuration Classes	所有自动配置	Auto-configuration classes provided by Spring Boot.
Test Auto-configuration Annotations	常见测试注解	Test-autoconfiguration annotations used to test slices of your application.
Executable Jars	可执行jar	Spring Boot's executable jars, their launchers, and their format.
Dependency Versions	所有场景依赖版本	Details of the dependencies that are managed by Spring Boot.

查看版本新特性；

https://github.com/spring-projects/spring-boot/wiki#release-notes <https://github.com/spring-projects/spring-boot/wiki#release-notes>



Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.

If you're looking for information about a specific version, or instructions about how to upgrade from an earlier release, check out the project release notes section on our wiki.

关注作者和知识库后续更新

尚硅谷
程序员标配，人手一套尚硅谷教程，自学一样...
已关注

SpringBoot2核心技术与响应式编程
基于SpringBoot2.3与2.4版本
关注

推荐阅读

05、Web开发
1、SpringMVC自动配置概览Spring Boot provides auto-configur...

03、了解自动配置原理
1、SpringBoot特点1.1、依赖管理父项目做依赖管理依赖管理 <...

02、SpringBoot2入门
1、系统要求Java 8 & 兼容java14 .Maven 3.3+idea 2019.1.21.1、...