Spring Memo

Kazuki Motohashi @ Konduit KK

Agenda

- Spring Framework
- Spring Boot
- Spring Cloud
- Spring Cloud Data Flow

Spring Framework

- provides patterns and structure
- handles common things that most developers need to do:
    - creates and manages singleton instances with application context
    - builds objects relation graph by dependency injection
    - provides database connectivity
    - web applications skeleton (Spring MVC)

Spring Boot

- starting point for building all Spring-based applications
- provides tons of architypes
    - REST API, WebSocket, web, streaming, tasks, …
    - security
    - support for SQL and NoSQL
    - Embedded runtime: Tomcat, Jetty, and Undertow
    - tracing, metrics, health status
- takes a lot of boilerplate codes and configurations away
- easy project initialization using Spring initializr

After initializing project with spring-boot-starter-web,
a web server works by only the following code

```java
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class ExampleApp {
    public static void main(String[] args) {
        SpringApplication.run(ExampleApp.class, args);
    }
}
```

Commonly used annotation

- @Autowired

- @Bean

for class

- @Conponent

- @Configuration

for service class

- @Service

- @Transactional

for controller class

- @RestController

- @Controller

Spring Cloud

- offers a simple programming model to common distributed system patterns
    - helps developers build resilient, reliable, coordinated applications
- **Service Discovery:**
    - load balancing and smart routing
- **Circuit Breaker:**
    - falt tolerance with a monitoring dashboard
- **Configuration Server:**
    - centralized configuration management
- **API Gateway**
- **Distributed Tracing**
- **OAuth2:**
    - single sign on and token
- **Consumer-Driven Contracts:**
    - service evolution patterns

Spring Cloud Data Flow

- management tool to define pipeline of components implemented using Spring Cloud Stream (streaming) and/or Spring Task (batch), deploy/undeploy applications, etc
- provides both CLI and GUI (based on Spring Flo)
- bash-like DSL (Domain Specific Language)

<div align="center">

http --port=8181 | log

</div>

  - communications between components are delt by messaging middleware
    - Kafka, RabbitMQ, Azure EventHubs, Amazon Kinesis Stream,…
- can be deployed at scale with:
    - (local), Apache Mesos, Kubernetes, Cloud Foundry