



Spring Boot Microservices

Beginner to Guru

Spring Framework and RESTful Services



Spring Framework and RESTful Services

- The Spring Framework has very robust support for creating and consuming RESTful Web Services
- Spring Framework has 3 *Distinct* web frameworks for creating RESTful services
- Spring Framework has 2 *Distinct* web client for consuming RESTful services
- There are also several popular libraries for creating and consuming RESTful services frequently used with Spring
 - Not covered in this course





Spring MVC - Web Framework

- Spring MVC is the oldest and most commonly used library for creating RESTful web services
- Part of the core Spring Framework
 - Compatible with Java EE (Jakarta EE in Spring 6)
- MVC - Model View Controller
- Has robust support for traditional Web Applications
- Based on traditional Java Servlet API
 - By nature this is blocking, non-reactive





Spring WebFlux - Web Framework

- Spring WebFlux was introduced with version 5 of the Spring Framework
- WebFlux uses project Reactor to provide reactive web services
 - Does not use Java Servlet API, thus is non-blocking
- Follows very closely to the configuration model of Spring MVC
 - Provides an easy transition for developers accustomed to traditional Spring MVC



WebFlux.fn - Web Framework

- Also introduced in Spring Framework 5
- WebFlux.fn is a functional programming model used to define endpoints
- Alternative to annotation based configuration
- Designed to rapidly and simply define microservice endpoints





Spring RestTemplate - Web Client

- RestTemplate is Spring's primary library for consuming RESTful web services
- Very mature - been a part of Spring for a very long time
- Highly configurable
- As of Spring Framework 5 RestTemplate is in maintenance mode
 - No new features are planned
 - Step before deprecation, Spring recommends using WebClient for new development





Spring WebClient - Web Client

- Spring WebClient was introduced in Spring Framework version 5
- This is Spring's reactive web client.
- By default uses Reactor Netty, a non-blocking HTTP Client library



Marshalling / Unmarshalling

- Converting Java POJOs to JSON or XML is called Marshalling
- Converting JSON or XML to Java Objects is called Unmarshalling
- By default Spring Boot configures Jackson to facilitate Marshalling and Unmarshalling
- Spring Boot does support several other libraries, however Jackson is by far the most popular
- Jackson will be the focus of the course





SPA - Single Page Applications

- RESTFul APIs are often combined with SPA applications for rich user web applications
- Popular client side SPA frameworks include Vue, BackboneJS, ReactJS, AngularJS, and EmberJS
- Frequent question is which framework is the “best” to use with Spring Boot
- The correct answer is that it does not matter
 - The framework used is decoupled from Spring via the HTTP / JSON (or XML) layer
 - All of these libraries can consume RESTful APIs
- Server side can be Spring Boot, .NET, Ruby on Rails, etc - the RESTful API abstracts the implementation



