DATE 2603 2

Spring Boot Web Services

★ Web Service →

- These are XML-based information exchange systems that use the internet for direct application-to-application Interaction. These systems can include programs, objects, messages or documents.
- It is a collection of open protocols and standards used for exchanging data b/w applications or systems.

→ Keys →

- Designed for Machine-to-machine (or app-to-app.) interaction.
- Should be interoperable - Not platform dependent.
- Should allow communication over a n/w.

→ Key terminology →

- Request & response
- Mssg Exchange Format
  - XML and JSON
- Service Provider or Server
- Service consumer or client
- Service definition.
- Transport
  - HTTP and MQ

→ Types →

① SOAP Web services → (Simple Object Access Protocol).

- It is a XML-based protocol for accessing web services. SOAP is a W3C recommendation for communication b/w 2 applications. It is XML based protocol. It is platform independent and language independent.

② RESTful web services → (Representational State Transfer)

- These are basically REST architecture based web services. In REST ~~services~~ Architecture everything is a resource.
- These are light weight, highly scalable and maintainable and are very commonly used to create API's and are very commonly used to create API's for web-based applications.

★ SOAP web services with Spring & spring Boot →

① Step-1 → Initialize a Spring web services application with spring boot.

② Step-2 → Overview of creating SOAP web service using contract first approach.

③ Step-3 → Define request & response XML structure

④ Step-4 → Define XML schema definition (XSD) for Request - GetCourseDetailRequest

⑤ Step-5 → Define XML schema definition (XSD) for Response - GetCourseDetailResponse

⑥ Step-6 → More about XML schema definition and implementing XSD best practices.

⑦ Step-7 → Introduction to Java API for XML binding (JAXB) and configuring JAXB 2.

⑧ Step-8 → Configuring an endpoint for GetCourseDetail Request

⑨ Step-9 → Spring web services configuration - Message Dispatcher Servlet

⑩ Step-10 → Spring web services configuration - Generating WSDL.

⑪ Step-11 → Using Wizdler to execute SOAP requests.

⑫ Step-12 → Implementing a service - Course Detail Service - backend within memo.

⑬ Step-13 → Implementing SOAP web service for GetAll CourseDetailsRequest

⑭ Step-14 → Quick introduction to diff. parts of a WSDL,

⑮ Step-15 → Implementing SOAP web service for DeleteCourse DetailsRequest

(16) Step-16 → Improving the Delete Course Details Request - using an Enum for status.

(17) Step-17 → Exception handling and SOAP fault responses.

(18) Step-18 → Implementing security for SOAP web services with ws security.

★ RESTful web services with Spring & Spring Boot →

(1) Step-1 → Initializing a RESTful services project with spring boot.

(2) Step-2 → Understanding the RESTful services we would create in this course.

(3) Step-3 → Creating a Hello World service.

(4) Step-4 → Enhancing the Hello World service to return a Bean.

(5) Step-5 → Quick review of Spring Boot Auto configuration and Dispatcher servlet.

(6) Step-6 → Enhancing the Hello world service with a path variable.

(7) Step-7 → Creating User Bean and User Service.

(8) Step-8 → Implementing GET Methods for User resource.

(9) Step-9 → Implementing POST Method ~~to~~ to create user resource.

(10) Step-10 → Enhancing POST Method to return correct HTTP Status code and location.

(11) Step-11 → Implementing Exception Handling - 404 Resource not found.

(12) Step-12 → Implementing generic exception Handling for all resources.

(13) Step-13 → Exercise : User post resource and exception handling.

(14) Step-14 → Implementing DELETE Method to delete a user resource

(15) Step-15 → Implementing validation for RESTful services.

(16) Step-16 → Implementing HATEOAS for RESTful services.

(17) Step-17 → Overview of advanced RESTful service features.

(18) Step-18 → Internationalization for RESTful services.

(19) Step-19 → Content negotiation - implementing support for XML.

(20) Step-20 → Configuring auto generation of swagger documenta

(21) Step-21 → Introduction to swagger documentation format.

(22) Step-22 → Enhancing swagger documentation with custom annotations.

(23) Step-23 → Monitoring API's with spring boot actuator.

(24) Step-24 → Implementing static filtering for RESTful service.

(25) Step-25 → Implementing dynamic filtering for RESTful service.

(26) Step-26 → Versioning REST ful services - Basic approach with URI's.

(27) Step-27 → Versioning RESTful services - Header and content negotiation approach.

(28) Step-28 → Implementing basic authentication with spring security.

DATE

★ Connecting RESTful web services to JPA →

(29) **Step-29** → Overview of connecting RESTful service to JPA.

(30) **Step-30** → Creating user entity and some test data.

(31) **Step-31** → Updating GET methods on user resource to use JPA.

(32) **Step-32** → Updating POST and DELETE methods on user resource to use JPA.

(33) **Step-33** → Creating Post entity and many to one relationship with user entity.

(34) **Step-34** → Implementing a GET service to retrieve all posts of a user.

(35) **Step-35** → Implementing a POST service to create a post for a user.

★ RESTful Web Services - Best Practices →

(36) **Step-36** → Richardson Maturity Model

(37) **Step-37** → RESTful Web Services - Best Practices.