Walmart Interview Questions (04/10/2018)

Client: (1 hour) (Webex Video interview)

* Tell me about yourself
* Develop a simple Spring application

App:getter/setter==Test:main/applicationcontext/App instance==XML/<beans><properties>

* Bean Lifecycle

@PostConstruct==@PreDestroy

* Explain @Bean, @Component annotations (Asked me to write code)

@Component is class-level Annotation(@Controller, @Service, @Repository) for implicit object injection

@Controller

public class LoginController

{ --code-- }

@Bean lets you configure your single bean explicitly.

@Bean

Integer theNumber(){

return new Integer(3456)}

* What other annotations did you use in Spring

@Required: to inforce parameter

@Autowired: implicitly injects the dependencies.

@@Component is class-level Annotation(@Controller, @Service, @Repository) for implicit object injection.

@Bean lets you configure your single bean explicitly.

* Which Nosql database did you use.

MongoDB/DaynmoDB

* Why Nosql database

No fix Schema, faster, scalable, performance oriented and can handle big and complex data.

* Rest Vs Soap

SOAP==Protocol, jax-ws, defines own standard & security, more bandwidth, XML only

Rest== state transfer, jax-rs, uses protocol security, less bandwidth, xml,json.html…

* Why Soap is more secure

Through wsdl uses own Standard and schema plus defines own security.

* Show me how can you Create & Consume REST API (Again asked me to code)

Publisher.java==

@Path("/hello")

**public** **class** Publisher {

@GET

@Produces(MediaType.***TEXT\_PLAIN***)

**public** String met() {

**return** "HhhMmmm";

}

Web.xml==

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* version=*"3.1"*>

<display-name>Restok</display-name>

<servlet>

<servlet-name>Jersey REST Service</servlet-name>

<servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>

<init-param>

<param-name>jersey.config.server.provider.packages</param-name>

<param-value>com.demo</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>Jersey REST Service</servlet-name>

<url-pattern>/rest/\*</url-pattern>

</servlet-mapping>

</web-app>

Consumer.java==

**public** **class** Client1 {

**public** **static** **void** main(String[] args) {

ClientConfig config = **new** ClientConfig();

Client client = ClientBuilder.*newClient*(config);

WebTarget target = client.target(*getBaseURI*());

System.***out***.println(target.path("rest").path("hello").request().accept(MediaType.***TEXT\_PLAIN***).get(String.**class**));}

**private** **static** URI getBaseURI() {

**return** UriBuilder.*fromUri*("http://localhost:8080/Restok").build();

* Spring Security / OAuth (How you handle authentication & authorization)
* Write implementation of OAuth (Configuration)
* Algorithm for Fibonacci Series.

**public static void main(String[] args) {**

**int n = 14, t1 = 0, t2 = 1;**

**for (int i = 1; i <= n; i++)**

**{**

**System.*out*.print(t1 + " + ");**

**int sum = t1 + t2;**

**t1 = t2;**

**t2 = sum;**

**}**

* CI/CD pipeline
* Spring Interceptors

preHandel()==before sending request to the controller

postHandel()==before sending responset to the client

* AWS – Elastic BeanStalk, EC2 Instance
* Java 8 – What features you used in your recent projects
* Functional Interfaces
* Explain Docker