++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

ADV JAVA

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

1)Servlet – ServletContext within the [ServletConfig](https://docs.oracle.com/javaee/7/api/javax/servlet/ServletConfig.html" \o "interface in javax.servlet) object, only one ServletContext object per web application, web.xml file using the **<context-param>**

**getContextPath()** **getInitParameter**([**String**](http://docs.oracle.com/javase/7/docs/api/java/lang/String.html?is-external=true) name) **removeAttribute**([**String**](http://docs.oracle.com/javase/7/docs/api/java/lang/String.html?is-external=true) name)

2)SpringBoot - @EnableAutoConfiguration @Configuration main Method Entry Method @EnableSpringbootApplication

@SpringApplication–

@SpringbootApplication

3)JDBC - checkpoints transcation

savePoint save the current sate of database which can be then rollback

CheckPoint while transaction is saved into db before this db is consistent state all transaction is commited

RollbackPoint CommitPoint

4)DriverManager Class - **synchronized void registerDriver(Driver driver) Connection getConnection(String url) throws SQLException**

5)RestFUl - without changing state of server on subsequent invocation Idempotence In the context of REST APIs, when making multiple identical requests has the same effect as making a single request independent of the number of times

Immutability Statefulness Uniformity

6) JPA entities – cascading mode **When we perform some action on the target entity, the same action will be applied to the associated entity**

7) testing spring boot entire context without starting server

@Test

@WebMvcTest focuses **only** on Spring MVC components

@SpringBootest entire container context

@SpringBootIntegrationTest

8)dynamic Interception of req,res = Servlet Container,

ServletConfig interface Container create one ServConfig Object this is argument to init method(servletConfig config) getInitParameter(String)<servlet-class, name, url-pattern,init-param, param-names,param-value

ServletContext, ServletFilter preprocessing postprocessing conversion logginh compression encryption validataion pluggable (init() doFilter() destroy())

FilterChain pass control to next filter responsible to invoke the next filter

9) POM file project object model configuration and diff dependencies

10) access resources in Service status codes WeB-INF/ - SC\_NOT\_FOUND SC\_INTERNAL\_SERVER\_ERROR ESC\_BAD\_REQUEST

11) JDBC drivers use of native protocol type 4 driver= native-protocal binary messeages into driver

pure java drivers, JDBC-Net, NATIVE-api, JDBC-ODBC Bridge plus ODBC driver

JDBC-ODBC thin driver function calls client side

Native-APi not written entirly in java client side

Network protocol driver- application middleware

Thin Driver – vendor-specific

12)createStatement() preapreStateMent() preapreCall() CallableStatement() stored procedures and functions nativeSQl() running for long time

13) caches in Hibernate Session Object 1level cache byDafult not available for entire application, 2level sessionFactory for entire application

Types : read-only ,nonStrict read-write, read-write, transcational

EH, YES YES YES NO

OS, YES Yes Yes No

Swarm, Yes Yes No No

JBoss No No No Yes

CacheConcurrencyStrategy.READ\_ONLY cache.provide\_class

, query cache only identifer value used conjuntion with 2nd level no state of of entity, Database cache

Cache mode controls how session interacts with 2nd level cache

Better 2nd level cache is EHCache

14)spring-boot-secure-actuators , start-web-services, dev-tools, starter-security

15) @WebServlet()

16) Request , Response interfaces

17)@JoinColumn

18) attributes of <jsp:useBean>… id scope class type beanName

19)criteria add addOrder setFirst setMax list setProjection API aggregate HAS\_A resultSet interface maintains cursor next prev Scrollable

20) automatic Config in spring boot – requesthandler retype() getName() dispatcher Process\_Input\_Data registry viewResolver render model in browser without view tech handlerMapping

21)@GetMappig ----

22)Scriplet <% ; %>

23)Embedded – JoinColumn

24)@Generator @GneratedValue @IdGenerator @Sequenece

25) SessionFactory thread safe? yes Second level cache

26) JSP lifecycle can override jspinit() jspDestory() but cannot over\_jspService()

27) Inheritance in hibernate Table per subclass, per class hierarchy, per object, per concrete class

28) Without auto-commit in connection executeUpdate execute executeQuery

29) Maven default lifecycle

30)getParameters()…………ServletRequest classes string[]

31) /books/{id}

32) test apis in webservice RestTemplate client like postman execute() delete(), RestManager runtime mapping of endpoints managedResource can be registed in /schema /config

33) Hibernate advs and diss

34) jspinit(), \_jspService(), jspdes()

35) @Entity @Key

36) @Aspect multi classes @JointPoint respresents method executions @Advice action at jointPoint

Types @Before before jointpoint @AfterReturn @AfterThrow @After finally @Around surrounds custom behaviour before and after most powerful @PointCut runs at any join point matched by pointcut @Around 37)lookup method injection prototype-singleton singleton-prototype

38)@Controller

39)hibernate-hdm2ddl.auto validate schema create-only drop update create cretae-drop none

40)jsp Action tag <forwar page> <params name value> </>

41) RestController

42) CommandLineRunnner can have multiple beans functionalInterfaces run()

43) accessing maven in cmd mvn site mvn clean java package

44)session.get(Class, ?) hit db and return actual obj null

Session.load() return proxy without hitting db temp object ObjectNotFound

45)scope in jsp default =page HttpJspPage this request response session application servletContext

46)Tags

47) web.xml http.xml, 404.html

48) doGet() doPost()

49) Optional tackle nullPointerEx orElse() isPresent()

50) <mapping=”” class=””>

51) Cookie = HttpSession, HttpCookie clone() getMaxAge() getName() getVersion()

52) CrubRepo

53) @Autowired <context:annotaion-config>

54) setAttribute()

55) Function level annotation

56) @ReqParam @PathParam @PathVariable

57) action directive declaration jsp <%@page import, contentType buffer> <%@include file><%@taglib>

58) application.properties app.name server.port app.admin.enabled servlet.context-path

59) isIDPresent, existsById isEntityPresent

60) call webservice from spring RestTemplate CallWSviaJSON CallWS

61) getPathInfo() uRL client sent when it made request

62) SessionListener maintain log details

63) GetResource : URL class

64) ServletEx, IllgalArgEx, UnavaliableEx, ServletContextEx

65)app.setAttribute() pageContext.set concretConnection()

<%=pageContext.getservletContext().getAttri

bute(“hitcount”)

66) <message-resources struts

67) .hbm hibernate.cfg.xml

68) ServletContextListener contextDestroyed contextInitialized ServletContextAttributeListener ServletRequestListener

69) Network Protocol driver Thin driver

70) Dirty object state checks changes in order to sync updated state

Lazy doesnot load parent object

Eager - load child automatically when parent is called

71) Log4j and Logback

72) hibenate entites can have transisent, detached, persistent Attached associated Detached associated before Persistent associated with object Transient never associated

Persistent object still exists memory until session.close()

73) app Context ClassPathXmlApplicationContext AnnotationApplicationContext WebApplicationContext WebPathXmlApplicationContext

74) Stereo Type Annotations @service @Repository @Controller

75) shut down of IoC container(XML) create configure assembles dependencies beanFactory and applicationContext @PreDestroy when been is removed when context shutting down registerShutdownHook()

76) Spring provides a factory javax.jws.WebService javax.jws.WebServiceProvider

@WebServiceProvider()

77) session.update() session.keep() session.saveOrUpdate() session.load() session.evict() - remove object and its collections from 1st level cache  
78) Spring’s core transaction management abstraction : TransactionManager PlatformTransactionManager PlatformManager PlatformTransaction

79)Native API partly JAVA DRIver

80) EJBContext access to the container provided runtime context of instance

Extended by sessionContext EnityContext MessageDrivenContext

1. Implicit Object In JSP JspWriter out don’t need HttpServletReq req, res ServletConfig config =initialization parameter in web.xml ServletContext app = only once by WC HttpSession session PageContext pageContext Object page Throwable exception
2. Degree of locking for sync static method sync block prevents multiple threads to access
3. Javabeans componenet architecture
4. beanInfo Class info about methods events props getPropertyDescriptor returns PropertyDescriptor[]
5. properties in beans <bean id=”bookD” class>context.getBean()</> @Bean
6. reflection getClass() getMethods(), serialization Object into byteStream writeObject() readObject() for ObjetOutputStream class need to be searlized class
7. IntroSpectors way tools to learn about properties event and methods we use low-level reflection find explicit then if not find classes superclass
8. Init(Config) service(req, res) destroy()
9. HttpServlet Class doPost() doGet() doTrace()
10. Translation type error
11. Registry storing and retrieving remote object bind() unBind() remote(), UnicastRemoteObject proxy stud classes rmic tool extends remoteServer exportObject()
12. Size of output buffer 512 bytes
13. ActionEvent to every EventListener paramString() getModifer() getActionCommand() getWhen()
14. Entity beans is business logic persistent vs session beans is used to implements services not persistence
15. Error page in jsp
16. OutputStream o = response.getOutputStream()
17. getHeaders()
18. taglib tag-class custom tag
19. tag interface doEndTag() doStartTag() getParent() release()

SKIP\_PAGE doEndTag()  
 EVAL\_BODY\_INCLUDE doStartTag()  
 SKIP\_BODY return of doStaartTag() and doRNtag()

EVAL\_PAGE – doEndTag()

SKIP\_PAGE\_AGAIN SKIP\_BODY\_AGAIN

1. Entity bean home interface
2. Value objects
3. Servlet getinitParameter()
4. Session bean stateless entity stateful
5. ejbCreate ejbStore decides entity beans data to the data-base
6. IllegalStateException
7. Object Mapping
8. Stub of object java.rmi.Remote java.io.Serializable
9. JDBC\_ODBC
10. bean-managed persistent you are responsible for sync must read and write database directly vs container-managed persistent no need to manage container handles
11. Servlet interface

* Init(ServletConfig config)
* Service(req, res)
* Destroy()
* getServletConfig()
* getServletInfo()

1. Hibernate

* Entities can be transient, detached = when entityManger closed , persistent = persistent can have memory until session.close() detached will be there = entityManagerFactory created by persistent
* Get() when don’t know if object exist or load()
* Primary key - generator class native
* Session not thread safe no method of remove() but session factory is
* Validate() return input if validation fails
* Interfaces -session, config, transaction, query and criteria
* Class - sessionfactory
* @version locking @basic default
* Date -- @mapkeytemporal
* Embedded type - @ElementCollection
* Enum - @MapKeyEnum
* @OrderBy - ordered

1. JDBC -

* Count - Result.getmetadata
* Register driver (4) and forName and getProperty, sun.jdbc obj
* Interfaces - connection statement result set
* Jdbc driver - type 4 (native protocol)
  + driver for internet - type 3 net protocol pure
  + thin driver - type 4 JDBC net pure java driver
  + native api partly - type 2
  + Odbc - type 1
* Drivermanager.getConnection -> connection pool - datasource => classnotfound => static vlock
* , concur\_read\_only
* Prepared statement - compiled
* Transaction isolation - READ\_UNCOMMITED 🡪 dirty reads non repeatable,
  + READ\_COMMITED 🡪 dirty reads prevented non repeatable
  + REPEATED\_READ 🡪 dirty read and non repeated prevented
  + SERIALIZABLE 🡪 dirty and non repeated and phantom prevented
* Result set = type\_forward\_only, Type\_scroll\_insensitive, Type\_scroll\_sensitive, not (READ\_ONLY, UPDATABLE)

1. JSP

* Jsp attribute no page not implicit object - context
* Jsp - HttpjspBase
* Jsp:useabean no name
* jspInit and jspDestroy (page transition) override no \_jspService override
* out jspwriter
* getOutputStram - IllegalstateException - runtime
* EL implicit object - Map

1. Spring

* Two dynanmic proxy - jDK ,CGLIB
* Spring provides - Resource management, exception, transaction, resource unwrapping (Not blob and clob abstraction)
* Auto inject 🡪 @Autowired @Qualifier
* If not property need to provide id else ref is sufficient
* @PostContrust before callback method
* @Configuration
* @Transctional(PROPOGATION\_REQUIRES\_NEW) - new transaction and run within own process, with transactional template class, id is transational manager
* JdbcTemplate does not provide data source access
* Resultset parsing and merging - resultsetextractor
* Rollback - for runtime
* Stateful bean - prototype scope
* Factory method - BeanCreationEx factory class, factory bean no error
* @autowiring -
  + No mode - default explicit name ref
  + Byname - property name
  + Bytype - property type
  + Constructor - same as bytype, fatal error
  + Autodetect - first by constructor if not bytype

1. AOP

* @Pointcut - public methods, find out jointpoints
* @Jointpoint - you can use aop
* Return advice - without throwing error
* Advice - action taken by aspect at joint point
* @After - once joint point finishes
* @AfterReturning - joint points returns or throws
* @Order - precedence
* @Intoduction - interfaces - @DeclareParents
* Weaving - target objects - ajc