

Please note, not all the answers are correct, in the notes you can find more details (something is wrong with this assessment):

- Question No. 1

How to define composite primary key for a given a class Employee with 2 primary key columns (emp no and department). Choose the right option

Class EmployeeId

@Id

private long empNo;

@Id

private String department;

@IdClass(Employee.class)

@Entity

Class Employee

private long empNo;

private String department;

Class EmployeeId

private long empNo;

private String department;

* @IdClass(EmployeeId.class)

@Entity

@Id

private long empNo;

@Id

private String department;

Notes: Need annotations, entity for class and id for fields

- Question No. 2

For getting value from the uri path in a controller we have to use the annotation

* @PathVariable

@UriVariable

@Value

@PathVariable

Notes: PathValue no doubt

- Question No. 3

By default rollback happens for @Transactional method, when

A checked exception is thrown

* An unchecked exception is thrown

Both A and B

No default rollback mechanism

?

Notes: n/a

- Question No. 4

Which of the following frameworks are supported in spring cloud circuit breaker?

Hystrix

Sentinel

Spring Retry

*. All of the above

Both A & C

Notes: Hystrix, Sentinel, and Spring Retry are supported frameworks (validate name is Sentinel and not Sentinal)

- Question No. 5

Which components, intercepts all request in Spring MVC ?

All of the above

ControllerServlet

FilterDispatcher

* DispatcherServlet

Notes: n/a

- Question No. 6

Which is true about the statement spring.jpa.hibernate.ddl-auto = create

During application start up, all the existing tables will get updated with new columns or constraints if any.

During application start up it will drop and create all the tables.

During application start up all the data in the existing tables will be lost.

Both B & C

* All of the above

Notes: validate if this is correct (Drops any existing schema and then creates a new one every time the application starts. This mode is commonly used during initial development stages where schema changes are frequent and preserving data is not crucial.)

- Question No. 7

Which of the following are the responsibility of an API gateway

Routing

Security

Monolith Strangling

* Bpth A & B

All of the above

Notes: n/a

- Question No. 8

How to add data to the info actuator?

Using info.* properties in application.properties

Implementing InfoContribution Bean

* Both A and B

None of the above

Notes: n/a

- Question No. 9

Which of the below statements are true regarding @Version in entity?

Each transaction that reads data from entity holds the value of the version property.

When a transaction makes an update it will check the version and throws OptimisticLockException exception if value has changed in between

For versioned entities optimistic locking is available by default

* All of the above

Both A & C

Notes: Need validation

- Question No. 10

Can we use JdbcTemplate with @Transactional ?

Run time error

Compilation error

No transaction will be created for the queries with JdbcTemplate

* Yes

Notes: n/a

- Question No. 11

Which is highest isolation level in spring transaction?

Read Comitted

Read Uncommitted

* Serializable

Repeatable Read

Notes: no doubt

- Question No. 12

In Spring @RestController is a

* Annotation

All of the above

Meta-Annotation

Stereotype

Notes: validate

- Question No. 13

While using spring cloud config, Where do we need to mention the server path in the client?

application.properties

application.yml

bootstrap.properties

* Both A & B

Notes: n/a

- Question No. 14

From which property eureka uses for registering client name

eureka.client.name

spring.cloud.client.name

eureka.application.name

* spring.application.name

Notes: n/a

- Question No. 15

How many profiles can be defined for a spring application

(2^{63})

(2^{63}) - 1

* Infinite

(2^{31}) - 1

Notes: there's no theoretical limit however $2^{31} - 1$ seems more accurate

- Question No. 16

Which bean scope creates an instance when ever the bean is requested

Prototype

* Request

Session

Application

Notes: n/a

- Question No. 17

Which of the following dependencies are injected by spring-boot-starter-test?

JUnit

* Hamcrest

JsonPath

Both A & C

All of the above

Notes: JUnit Jupiter, Hamcrest, and Mockito are the dependencies added by test starter, however here is mention Junit with a "u" in small caps, therefore maybe is only Hamcrest.

- Question No. 18

Which method can be used to retrieve a JDBC template while using JdbcDaoSupport?

GetSpringJdbcTemplate()

getJdbcTemplate()

* getJdbcTemplate()
getTemplate()
Notes: n/a

- Question No. 19

Which of the below code will fetch all users from UserRepository?

userRepository.getAll();
* userRepository.findAll();
userRepository.find();"
Notes: n/a

- Question No. 20

Which of the following is the proper syntax for overriding a run method in a CommandLineRunner?

@Override
void run(String l args) 0
@Override
void run(String... args) 0
@Override
• void run (String l args)
throws Exception t
* @Override
void run (String... args)
throws Exception 0
Notes: public void run(String...args) throws Exception {

- Question No. 21

Which of the follwing is true for spanid

SpanID will be unique for a microservice
spanId will be unique for a request to an API
All related requests generated from a same request will share same spanId
* Both B & C
Notes: n/a

- Question No. 22

Which of the following is not a stereotype in spring?

Service
Component

* Bean
Controller
Notes: n/a

- Question No. 23

Which are the features of a REST specification

Idempotence

* Stateless

Both A & B

None of the options

Notes: n/a

- Question No. 24

Which of the following code is the right way of creating a bean?

@Configuration

Class DataConfig {

@Bean

DataSource myDataSource() {

@Component

Class DataConfig {

@Bean

DataSource myDataSource() {

Class DataConfig {

@Bean

DataSource myDataSource() {

* All of the above

Both A & B

Notes: this are the 3 correct ways: @Configuration

public class AppConfig {

@Bean

public MyServiceImpl myService() {

<beans><bean id="myService"

class="com.acme.services.MyServiceImpl"/></beans>

- Question No. 25

What will after advice do in Spring AOP?

* Advice will be executed when a join point exits normally or by throwing an

exception.

Advice will be executed only when a join point exits throwing an exception.

Advice will be executed only when a join point exits normally.

None of the above

Notes: After advice, declared by using the @After annotation, is executed after a matched method's execution, whether or not an exception was thrown. In some ways, it is similar to a finally block. In case you need advice to be triggered only after normal execution, you should use the returning advice declared by @AfterReturning annotation. If you want your advice to be triggered only when the target method throws an exception, you should use throwing advice, declared by using the @AfterThrowing annotation.

- Question No. 26

What will be the result of the below code?

```
@Value("dummy")
```

```
public String findValue(String a, String b){
```

```
    Run time exception
```

```
    Compilation error
```

```
    When called returns the string "dummy"
```

```
    * execute the method with the value of a and b set to "dummy"
```

Notes: Validate, nothing happens actually

- Question No. 27

Java Config can be applied on final classes

TRUE

* FALSE

Notes: Validate

- Question No. 28

In Spring @Transactional can be used on top of

Class

Interface

Method

Both A & C

* All of the above

Notes: Validate

- Question No. 29

Which of the following cannot be the return type of the request method in a controller

* Object

String

ModelAndView

Void

Notes: Validate

- Question No. 30

Which annotation initializes a spring cloud config server?

@EnableConfig

* @EnableConfigServer

@EnableSpringConfigServer

@EnableSpringConfig

Notes: Validate

- Question No. 31

Which are the default methods in a repository?

findAll()

save()

saveAll()

* Both A & B

All of the above

Notes: Validate

- Question No. 32

For a JUnit test with @ExtendWith(SpringExtension.class), If you annotate your @Test with @Transaction, then which of the following are true

* Transaction is rolled back when there is a unchecked exception

Transaction is rolled back when there is a checked exception

Transaction is rolled back always

Both A & B

Notes: Validate

- Question No. 33

Which annotation helps you in setting up a eureka client in spring

cloud

@EurekaClient.

@EnableEurekaClient

* @EnableDiscoveryClient

Both B & C

Notes: Despite EurekaClient is valid now is deprecated

- Question No. 34

How to enable authentication for eureka server?

* Using Spring Security

Eureka can't be enabled with authentication

Using eureka's built in security system

Both A & C

Notes: Eureka doesn't have security so spring security is used, validate

- Question No. 35

Which ACID principle refers to transaction commit

Availability

Consistency

Integrity

* Durability

Notes: n/a

- Question No. 36

How to apply lombok changes directly to the source files rather than the class files

Using Delombok

Not possible

Using eclipse plugin

* Using @Generated annotation

Notes: Validate

- Question No. 37

If we want to expose a container port 3000 to local port 8080 which is the correct docker compose format?

* ports:

8080:3000

Notes: validate

- Question No. 38

In Spring data for each transaction, the commit method defined as per___ interface is evoked?

Platform TransactionManager

* TransactionManager

Spring TransactionManager

DataTransactionManager

Notes: validate

- Question No. 39

Docker instructions are saved in which file?

* Dockerfile

.dockerfile

Notes: n/a

- Question No. 40

Which of the following are the primary approaches for sagaco ordination

Choreography

Voting

Orchestration

All of the above

* Both A & C

Notes: validate

=====

- Question No. 1

Which of the following statements regarding spring data jpa and hibernate are true?

* You can use hibernate inside spring data

Spring data jpa is a jpa implementation

Hibernate is not a JPA provider

All of the above

Both A & C

Notes: validate

- Question No. 3

Identify the method name that should be replaced by in the below code?

```
HttpEntity request = new HttpEntity<>(new SaveUserDTO("100",  
"Alex");
```

```
    URI loc = restTemplate("/user", request);
```

```
postForLocation
```

```
exchange
```

```
postForURI
```

```
* postForObject
```

Notes: validate

- Question No. 4

Two possible problems that can occur if we are not solving a cross cutting concern via AOP?

None of the above

* Code tangling, Code scattering

Code scattering, Increased execution time

Code tangling, Increased execution time

Notes: validate

- Question No. 6

Which of the following code snippet is the right implementation of ehcache?

```
@Component
```

```
public class SimpleBookRepository implements BookRepository {
```

```
@Override
```

```
@Cacheable(cacheNames = "books" , key="#isbn"
```

```
', condition="#realtime == false")
```

```
public Book getByIsbn(boolean realtime) {
```

```
return new Book(isbn, "Some book");
```

```
@Component
```

```
public class SimpleBookRepository implements BookRepository {
```

```
@Override
```

```
@Cacheable(cacheNames = ""books"
```

```
, key=""isbn"", condition=""realtime == false")
```

```
public Book getByIsbn(boolean realtime) {
```

```
return new Book(isbn, "Some book");
@Component
public class SimpleBookRepository implements BookRepository {
```

```
@Override
@Cacheable(cacheNames = "books" , key=":isbn"
, condition="realtime == false")
public Book getByIsbn(String isbn, boolean realtime) {
return new Book(isbn, "Some book");
...
}
```

Notes: @Cacheable(
 value = "squareCache",
 key = "#number",
 condition = "#number==10")

- Question No. 7

Which of the following is the proper implementation of MapStruct library?

```
@MapStruct
public interface UserObjMapper {
    UserMapper INSTANCE = Mappers.getMapper (UserObjMapper.class );
}
```

```
@Mapping
UserDto toDTO(User user);
@Mapper
public interface UserObjMapper {
    UserMapper INSTANCE = Mappers.getMapper (UserObjMapper.class );
}
```

```
@Mapping
UserDto toDTO(User user);
@Mapping
public interface UserObjMapper {
    UserMapper INSTANCE = Mappers.getMapper (UserObjMapper.class );
}
```

```
@Mapper
UserDto toDTO(User user);
@MapStruct
public interface UserObjMapper {
```

```
UserMapper INSTANCE = Mappers.getMapper( UserObjMapper.class );
```

...

Notes: @Mapper(componentModel = "spring")

```
public abstract class SimpleDestinationMapperUsingInjectedService {
```

```
    @Autowired
```

```
    protected SimpleService simpleService;
```

```
    @Mapping(target = "name", expression =
```

```
    "java(simpleService.enrichName(source.getName()))")
```

```
    public abstract SimpleDestination sourceToDestination(SimpleSource  
source);
```

- Question No. 8

What is the difference between the @Controller annotation and @RestController annotation?

There is no difference

* @RestController annotation combines @Controller annotation and @ResponseBody annotation

@RestController automatically adds REST based validations to the controller

@RestController adds validation based on REST verbs

Notes: n/a

- Question No. 10

Which bean helps you to route request in Spring Cloud gateway?

RouteMapper

Route

* RouteLocator

RouteConfigure

Notes: validate

- Question No. 12

For the given Entities below Choose the right method for the entity's repository fetch a record by roll number and name

```
@Embeddable
```

```
class StudentId{
```

```
private String rollNo;  
private String name;  
class StudentMarksf  
@EmbeddedId  
private StudentId id;  
private long mark;  
findByld (33, "Anu");  
* findByRollNoAndName(33,"Anu");  
findWithdRollNoAndldName(33,"anu");  
findByldRollNoAndldName(33,"anu");  
Notes: I.e. findByNameOrAgeAndActive
```

- Question No. 15

Can we use @Before advice to prevent execution of a method?

* No

Yes

Notes: Before Advice runs before target method execution; It cannot stop execution of the target method,

- Question No. 16

While using Read Committed isolation level in a transaction, which all challenges are solved

Phantom Read

* Dirty Read

Non Repeatable Read

All of the above

Notes: validate

- Question No. 17

In docker compose, which key helps you to specify service dependency?

* depends_on

dependsOn

depends

Notes: n/a

- Question No. 18

Spring cloud sleuth is using for

Data Management

Authentication

Security

Distributed tracing

Notes: n/a

- Question No. 21

What is the default scope of bean in spring

None of the above

* Singleton

Session

Prototype

Notes: validate

- Question No. 22

If JDBC Template is used without transaction, which of the following cases are correct?

* Connection is acquired and released for every method call

Calls DatabaseUtils to get the connection

TransactionSynchronizationManager is executed

Both A & B

Notes: validate

- Question No. 23

Which is not a feature of IOC?

Lifecycle Management

* Transaction Management

Dependency Injection

Object creation

Notes: validate

- Question No. 24

In Spring @Transactional is a

* Concern

Service

Advice

Cross Cutting Concern

Notes: validate

- Question No. 25

In docker compose, if we need to restart a service incase of failure, which property should be configured

reboot

docker-compose won't support restarting on containers due to failure.

* restart

restart-on-failure

Notes: validate

- Question No. 26

What does REST stands for?

Response State Transfer

Request Transfer

Request State Transfer

* Representational State Transfer

Notes: n/a

- Question No. 27

Which of the following are valid docker build commands

mvn clean docker:build

* docker build .

mvn clean build

Both A & B

Notes: validate

- Question No. 28

Which REST verb is used to update specific part of an entity?

GET

PUT

* PATCH

POST

Notes: n/a

- Question No. 31

Which property should specify the Spring Cloud server service name in the client's properties

* spring.cloud.config.discovery.service-id

spring.config.discovery.service-id

spring.config.serviceid

spring.config.discovery.serviceid

Notes: spring.cloud.config.discovery.serviceId

- Question No. 32

How to change the LAZY fetch type of an entity for a query to EAGER?

Using @EntityGraph

Using @NamedEntityGraph

* Not possible

Both A & B

Notes: validate

- Question No. 33

Which of the following are not a feature of SpEL

None of the above

Invoking a bean method

Access static fields of classes

* Reading environment variables

Notes: possible invoke static methods but not sure about fields, i.e.

@Value("#{T(java.lang.Math).max(appConfig.minValue, 100)}")

- Question No. 35

Which of the following code will set the system properties "app.name" to the variable appName?

@Value("#{app.name}")

private String appName;

@Value("#{systemProperties.app.name}")

private String appName;

* @Value("#{systemProperties["app.name"]}")

private String appName;

@Value("\${systemProperties["app.name"]}")

private String appName;

Notes: I.e. @Value("#{systemProperties['priority']}")

- Question No. 37

Which are the valid spring data repositories?

CrudRepository

JpaRepository

PagingAndSortingRepository

Both A & B

* All of the above

Notes: n/a

- Question No. 40

How does the detach operation affect the Entity object in 'new' state?

An exception will be thrown out

The operation is ignored

* The status of Entity will become detached.

None of the above

Notes: validate