1. Using Annotations, how could you annotate a JPA EntityManager of a Spring managed bean to provide Transactionally based interactions with a database? 1 / 1 point

- ○ @PersistenceUnit
- ○ @Inject
- ○ @Value("#{entityManager}")
- ◉ @PersistenceContext

Correct
Correct; This injects an EntityManager

2. How would you provide an implementation of the Spring data JpaRepository to? inject into your Spring managed beans? Pick all that apply. 1 / 1 point

- ☑ Provide the maven dependency for spring-boot-starter-data-jpa in your pom

Correct
Correct; In a Spring Boot application, proxying of a Repository is achieved by simply added the maven dependency

- ☐ Create a class that extends JpaRespository and provide implementations of its methods
- ☐ Provide a class that extends EntityManager an inherit its methods
- ☑ Create an interface that extends JpaRepository

Correct
Correct; Providing the interface will enable auto proxying to implement the interface

3. How do you Annotate a Repository to be a Restful service itself where json collections is wrapped in the element "course" and its url is "/courses? 1 / 1 point

- ○ @RestController
- ◉ @RepositoryRestResource(collectionResourceRel = "course", path = "courses")
- ○ @RepositoryRestResource(collectionResourceRel = "course")
- ○ @RepositoryRestResource

Correct
Correct; the attribute collectionResourceRel provides the element for wrapping json collections and the path attribute identifies the url

4. How would you define a query method in a JpaRepository<Student, Long> to get all students by their firstName and lastName by equality and return a Collection of Students 1 / 1 point

- ○ Collection<Student> getByFirstNameAndlastName(String s1, String s2);
- ○ Collection<Student> findByFirstNameEqualAndlastNameEqual(String s1, String s2);
- ◉ Collection<Student> findByFirstNameAndlastName(String s1, String s2);
- ○ Collection<Student> findByFirstAndLast(String s1, String s2);

Correct
Correct; the proxying mechanism for Repositories uses the findBy to generate a criteria Query based of the Repository entity. In this case it looks for the firstName and lastName properties of Student to create the query

5. What is an Excerpt with respect to Exposing a Spring Data Repository as a service? Select all that apply. 1 / 1 point

- [x] Excerpts can make use of SPEL expressions for calculated fields in their associated Projection

Correct
Correct; Using SPEL expressions we can inject accessible data into the projection and hence the displayed Excerpt

- [x] Is associated with a Projection

Correct
Correct; The Excerpt identifies the Projection it is based on

- [x] Is a Projection representation associated with a Collection and only the Projection properties are displayed when requesting a Collection of the Entity

Correct
Correct; Collections using an Excerpt display the Projection of that Entity not the Entity itself

- [x] A subset of an Entity in a Collection of Entities returned from a RestService

Correct
Correct; It is based of a Projection that defines a subset of an entities properties

6. What is @Query used for in a Repository Interface? 1 / 1 point

- [x] Using in tandem with @Modifying and @Transactional the query can be written as a Select for Update

Correct
Correct; @Modyfying indicates the Query for update and @Transactional will commit the result of the JPQL statement

- [x] Using it means that there is no restriction on the naming of a method that it annotates unlike finder methods

Correct
Correct; the method can be called anything that is being annotated as the @Query is defining the query not the method signature as in finder methods

- [x] Defines a JPQL or Native Query

Correct
Correct; The enclosed JPQL or SQL defines the Query

- [x] Using @Param you can match JPQL parameters to the interface method arguments

Correct
Correct; JPQL arguments can be named and cross referenced with method arguments annotated with @Param(JPQL parameter name)

7. What annotation defines a class as a subset of a JPA Entity Resource when exposing a JpaRepository through REST — 1 / 1 point

- ○ @Controller
- ◉ @Projection
- ○ @Entity
- ○ @Component

Correct
True, this annotation provides a name attribute and the entities that it provides a subset of properties to

8. What JPA annotation defines Cardinality between two entities of one entity containing a Collection of another Entity — 1 / 1 point

- ○ @ManyToOne
- ○ @ManyToMany
- ○ @OneToOne
- ◉ @OneToMany

Correct
True this is the one to many relationship defined

9. What's true about a “Hypermedia As The Engine Of Application State” (HATEOAS) Select all that apply — 1 / 1 point

- ☑ Embedded links in response json represent further actions that can be taken by the client

Correct
True Urls represent actions in the response

- ☐ A consumer of HATEOS responses must hard code all action urls themselves
- ☑ Allows you to not only send the data but also specify the related actions

Correct
True, this is the basis of HATEOS, action urls are provided in the response so the client does not have to code them themselves

- ☐ It is another name for HTTP

10. What does the configuration property spring.jpa.properties.hibernate.enable_lazy_load_no_trans=true do — 1 / 1 point

- ○ Commits transactions to the database and therefore state of the PersistenceContext
- ○ Flushes entities to the database
- ◉ Allows fetching of lazy entities through Hibernate by opening a temporary session
- ○ Hibernate Fetches all JPA Entities eagerly by default

Correct
True One to Many relationships in JPA fetch Collections (the many end) lazily, to avoid unnecessary trips to the database, this needs a session opened to connect to the database which is normally closed after a parent entity is retrieved in a one to many relationship