Quiz 2

1. **What are the methods for Rest service and what 's the meaning of them? CURD methods including:**

   Get: Retrieve/Read Data

   Put: Update data

   Post: Add new data

   Delete: Delete existing data

2. **What is the difference between controller and rest controller?**

   @RestController = @Controller + @ResponseBody

   if we need to return json data, we use @RestController, otherwise, if we need to return jsp/html data, we use @Controller.

3. **How do you make sure your REST service can return both JSON and XML format data?**

   produces = ({ "application/xml" , "application/json" })

4. **Explain what cookie and session is and when to use which one.**

   Both cookie and session are used to store data.

   Cookie is for front-end storage; size is small and it can be used with every http request

   Session is for back-end storage; bigger size and it has an expiration time. It will expire if time is out without users interaction or the tab is closed.

5. **What is CORS issue and how do you resolve it?**

CORS is cross origin resource sharing. An individual running front-end and an individual running back-end are not on the same domain would raise a CORS issue. We can resolve this by using WebMvcConfigurer:

use @EnableWebMvc and override addCorsMappings to add domains, methods or headers should be allowed by using allowedOrigins(), allowedMethods() and allowedHeaders().

6. **List the meaning of http error code: 400, 404, 500**

   400: User sent invalid request – syntax or parameter

   404: User sent wrong request URL

   500: Server has internal configure error

7. **What is the difference between ApplicationContext and BeanFactory?**

   Both them are containers and interfaces, but ApplicationContext extends BeanFactory and contains more features, like Convenient MessageSource access for i18n(internationalization).

8. List the scopes of a spring bean and the meaning of each.

singleton	scopes a single bean definition to a single object instance for each Spring IoC container
prototype	scopes a single bean definition to any number of object instance
request	scopes a single bean definition to the lifecycle of a single HTTP request
session	scopes a single bean definition to the lifecycle of a single HTTP session
application	scopes a single bean definition to the lifecycle of a ServletContext
websocket	scopes a single bean definition to the lifecycle of a websocket

9. **What is the 4 properties in a “data source” configuration?**

   driver, url, username, password

10. **Suppose there are two instances of “EmployeeUtils” in spring bean definitions, one is “publicUtil” and the other one is “privateUtil”. How to inject the “publicUtil” to controller?**

    @Autowired

    @Qualifier(“publicUtil”)

11. **What are the types of dependency injection?**

    Constructor-based: injection happened at constructing time

    Setter-based： injection happened when be used

    Field-based：based on Java reflection

12. **What are the annotations to use on Service and DAO class so that Spring can recognize them?**

    @Service, @Repository

13. **How do you create a brand new spring boot project that can do rest service and use jpa to connect to database? Describe the steps.**

    1） go to website: spring initializer

    2） choose spring version, maven, group id, artifiont id

3） add dependencies: web, jpa

4） click the zip button and save zip file to local

5） unzip the project and import it into IntelliJ

6） when using jpa to connect, just use annotation

14. **Following Problem 13, in the spring boot project, You are given URL: http://catalog-es-service.prod.walmart.com/es-service/product/0212/search?sourceId=all&logloc=stack011 in which, 0212 is product sequence, sourceId is product source, logloc is production location**

Write a controller ("ItemViewController") and retrieve the product information. Suppose the product class and service class have been created for you. Add all annotations.

```
@RestController

public class MainController {

    @GetMapping(value=" /es- service/product/{sequence}/search" )

    public Product ItemViewController(@PathVariable int swquence,

                                      @RequestParam(name=" sourceId" ) int id,

                                      @RequestParam(name=" logloc" ) String loc) {

        Product p = service.getProduct(id, loc);

        retrun p;

    }

}
```

15. **Following Problem 14, Write the service(ItemViewService) class and delegate the logic to DAO (ItemDao) class method. You will need to inject the Dao class to this service. Add all annotations.**

```
@Service
public class ItemViewService {
    @Autowired
    private ItemDao itemDao;
    public ItemDao itemDao() {
        return new ItemDao();
    }
}
```

16. **Following Problem 15, suppose you are injecting a helper(ItemViewHelper) class into the service class. Use XML and annotation separately to define and inject the helper class into service class. Do not mix XML and annotation.**

```
public class HelperManagement {
    @Bean
 public ItemViewHelper helper() {
     return new ItemViewHelper();
    }
}
```

```
<bean id=" helper" name=" helperBean"

      class=" com.service.

      ItemViewHelper" >

</bean>

<alias name=" helper" alias=" Helper" ></alias>
```

17. In spring boot application.properties, there is an entry defined as: walmart.es.service=financial. Write a complete code to get the value “financial” in service class.

```
@Autowired
Environment env;

public void getFinalcial() {
    String financial = env.getProperty( “walmart.es.service” );
}
```