How to Setup/Code a Spring Boot Starter App with Spring Data

1. Set up access to the data base manager and be sure the table(s) are defined using your DBMS access tool (pgAdmin (PostgreSQL or SSMS (Sql Server)):
    a. Create/Verify a login user (if you already have a user and password, skip this step):
        i. To Create: right-click the server name
        ii. Select Create ---> Login/Group Role
            1. General Tab: Enter username
            2. Definition Tab: Enter a password you will remember (same as user name for testing)
            3. Privileges Tab: Turn on Can Login
                Turn on Make it a Super User
            4. Click Save
        iii. You should see the user in the Login/Group roles list under the server
        iv. To verify the user name and/or change password
            1. Right click the username and click through the tabs

    b. Verify the database and table(s) in the data base you want to access
        i. Right click in the DataBase on the left-side
        ii. Open up the database name (click on >)
        iii. Open up the Schemas (click on >)
        iv. Open up Tables (click on >)
        v. You should see a list of tables in the data base
        vi. Make sure the table you want to access is in the list

        Note: if don't have a data base, create one:
            a. Right click on DataBases under the server name
            b. Click Create
            c. Give the data base a name
            d. Click Save

        Note: if the table(s) don't exist in the data base, create them:
            a. Right click on data base name under data bases
            b. Choose Query Tool
            c. Open the file to create the tables or write the SQL yourself
            d. Run the SQL to create the tables(s)
            e. Save the SQL to a file if you wrote it yourself

    c. Verify the column names and data types of the columns in the table
        i. Right click the table name
        ii. Select "Properties"
        iii. Click the "Columns" tab to see the column names and attributes
        iv. Click the "Constraints" tab to if there are any constraints
        v. Click through the type of constraints tabs and note which ones exist

How to Setup/Code a Spring Boot Starter App with Spring Data

2. Open the Spring Boot starter project:
    a. Verify there are packages for **controllers**, **daos** and **model** under the server package.
    b. If any of the packages are missing, create them (**Be sure they are under the server package!)**
3. Verify the Maven dependencies for Spring Boot and Spring Data (refer to the sample given in your class notes).
4. Verify the entries in the **application.properties** file:
    a. Connection string for the server has the correct:
        i. Server URL (http://localhost:#### *(#### = port number*)
        ii. Port number in the URL matches the data base server port
        iii. Look at server connection properties in the server interactive interface tool to find the server name and port
    b. Credentials for accessing the data base server (Username and Password)
5. Create a POJO in the ***model*** package for each table you want to access:
    a. The POJO name typically matches the table name it represents.
    b. Be sure you have these annotations before the class header:
        i. **@Entity** to the data framework this is a POJO for a table
        ii. **@Table** to the data framework the name of the table in the database
    c. Be sure the data types you assign the data members is compatible with the data type of the column in the table it represents.
    d. Be sure to have the definition of any data members the way you want them before having IDE generate any code.
    e. Code/Generate the methods for the POJO
    f. Add an **@ID** before the variable in the POJO that represents the primary key
    g. Add a **@GeneratedValue** if the value for the column is automatically generated by the data base manager
    h. Add **@Column** before any variable named differently that the column it represents.
    i. Add **@Version** before the version data member, if you have one
6. In the ***DAO*** package, code the objects necessary to create a Spring Data DAO service:
    a. Three things are required for the service to access data using a Data Framework:
        i. **Interface** – List of method signatures for methods available to applications
        Minimally the CRUD methods:
            - Get all objects
            - Get an object by primary key
            - Add an object
            - Update an object
            - Delete an object by primary key

How to Setup/Code a Spring Boot Starter App with Spring Data

ii. **Repository** - **interface** for gaining access to the Data Framework method

1. *extends* **JpaRepository** class sending it the class of the object to be accessed and the data type of primary key
2. Code methods that require custom SQL for the application using **@Query** and **@Param**

iii. **Implementation** - Code for the methods defined in the interface

1. Implements the methods for service interface (*have the IDE generate initial code for methods by clicking "implement methods" when you place the cursor on the long red squiggle under the class header.)*
2. Add **@Service before the class header**.
3. Code a reference to a Repository class object.
4. Code/Generate a constructor that take the Repository reference as a parameter as a parameter to the constructor so it will be dependency injected by Spring.
5. Add **@Autowired** before the constructor
6. Use repository methods to access the database such as:

   .save()
   .findAll()
   .delete
   .findById()

   (type repository object **name.** to see them all)
7. Consider adding **@Transactional** before methods add/change/delete the data base

7. In the ***controller*** package, code the controller class and its methods:
   a. Create a new class for the controller methods.
   b. Include the **@RestController** annotation before the class header
   c. Code **@RequestMapping** annotation if you want a prefix to be used on all URL paths,
   d. Define a reference to the Relational Data DAO service Implementation.
   e. Code the reference as parameter to the constructor so it will be dependency injected by Spring
   f. For each controller:
      i. Determine:
         1. Determine what the controller will do
         2. The DAO implementation method that will return the data you need.
         3. What will be returned from the controller (usually whatever the DAO method returns)
         4. What data the controller need to pass to the DAO method and where to get it.
         5. URL path the client will use to run the controller.
         6. The HTTP method required to run the controller.
      ii. Code the controller method to call the Implementation methods and return expected result using the information gathered above.

How to Setup/Code a Spring Boot Starter App with Spring Data

<u>Use the incremental development process:</u>

Once you have the POJO, Interface and Repository done, consider writing and testing one implementation methods and the associated controllers methods at a time.

i.e. Don't try to write all implementation methods and then all controllers.