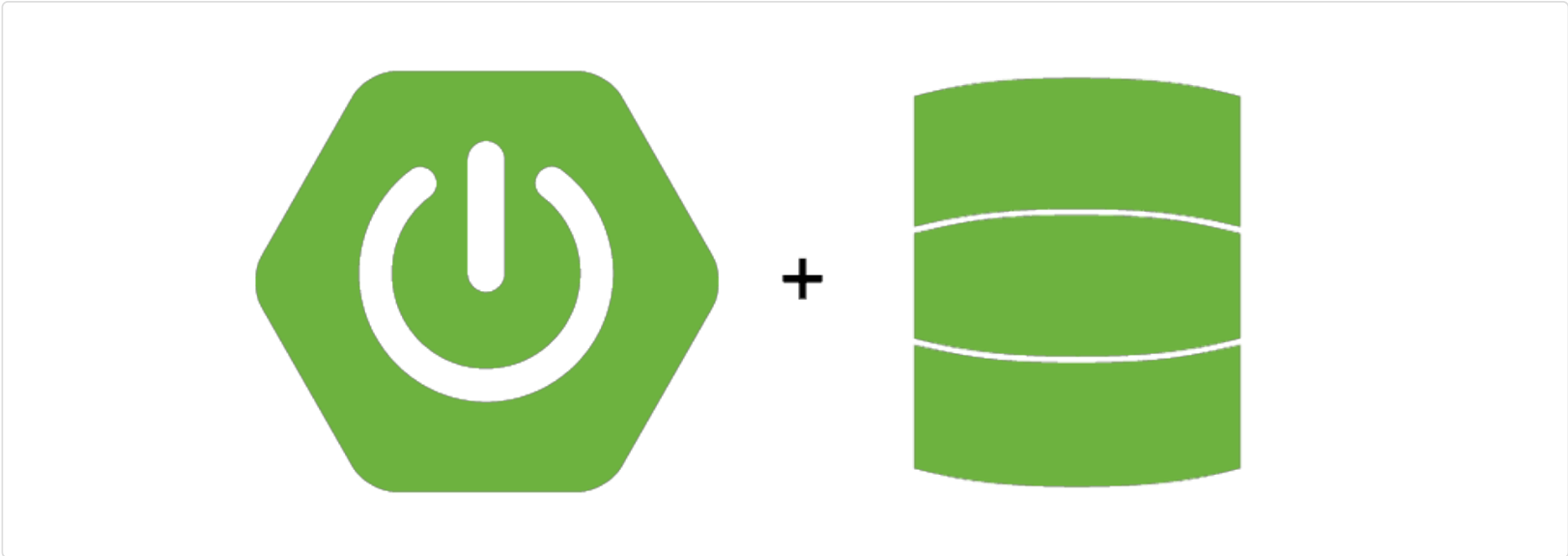


Spring Boot + Spring data JPA

By [mkyong](#) | March 22, 2019 | Viewed : 5,549 | +901 pv/w



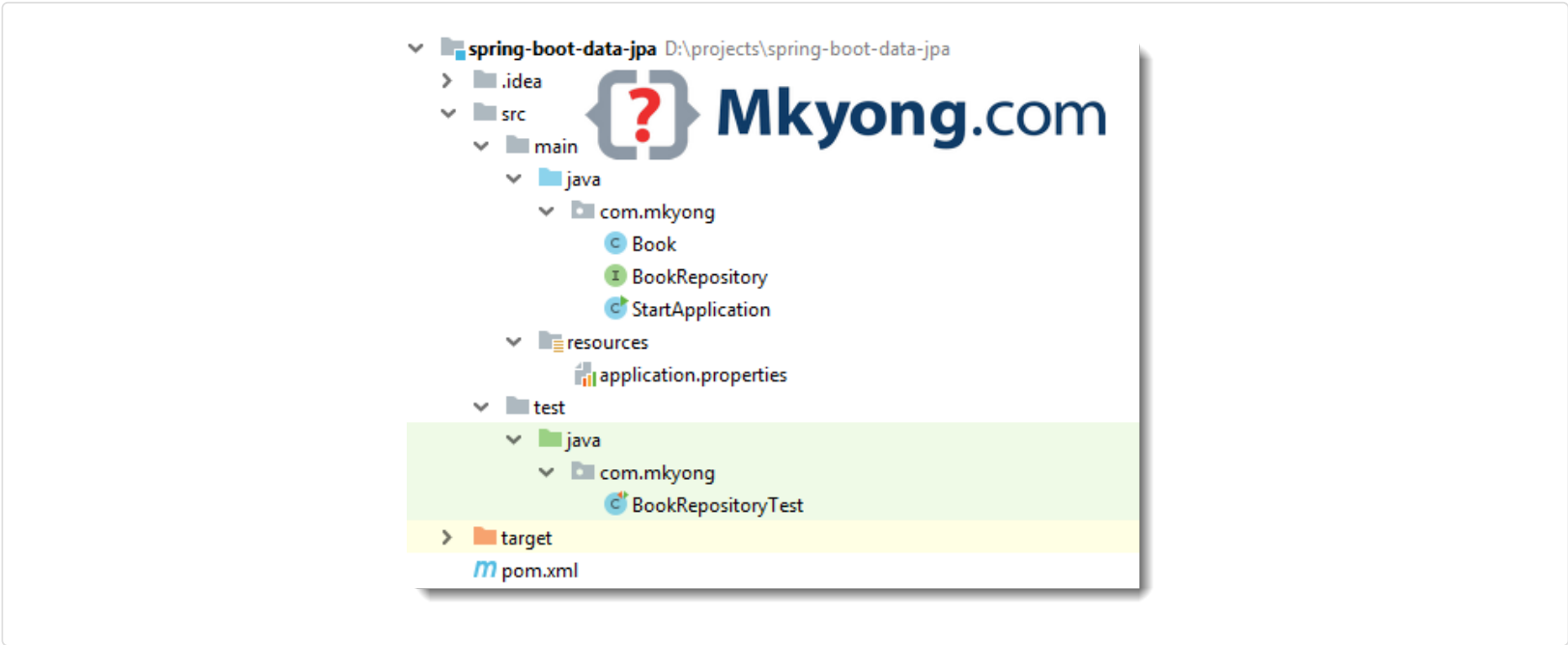
In this tutorial, we will show you how to use Spring Boot + Spring data JPA to save data into an H2 in-memory database and how to also query the data.

Technologies used :

- Spring Boot 2.1.2.RELEASE
- Spring 5.1.4.RELEASE
- Hibernate 5.3.7
- HikariCP 3.2.0
- H2 in-memory database 1.4.197
- Maven 3
- Java 8

Note
By default, Spring Boot 2 uses [HikariCP](#) as the database connection pool. Read this [Spring Boot algorithm](#) to choose a pool implementation.

1. Project Directory



2. Maven

Puts `spring-boot-starter-data-jpa`, it will get Spring data, Hibernate, HikariCP and all databases related dependencies.

```
pom.xml
```

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <artifactId>spring-data-jpa</artifactId>
  <packaging>jar</packaging>
  <name>Spring Boot Spring Data JPA</name>
  <version>1.0</version>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.2.RELEASE</version>
  </parent>

  <properties>
    <java.version>1.8</java.version>
    <downloadSources>true</downloadSources>
    <downloadJavadocs>true</downloadJavadocs>
  </properties>

  <dependencies>

    <!-- jpa, crud repository -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <!-- in-memory database -->
    <dependency>
      <groupId>com.h2database</groupId>
      <artifactId>h2</artifactId>
    </dependency>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>

  </dependencies>

  <build>
    <plugins>

      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>

      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>2.22.0</version>
      </plugin>

    </plugins>

  </build>
</project>

```

Display the project dependencies.

```
$ mvn dependency:tree

[INFO] org.springframework.boot:spring-boot-data-jpa:jar:1.0
[INFO] +- org.springframework.boot:spring-boot-starter-data-jpa:jar:2.1.2.RELEASE:compile
[INFO] | +- org.springframework.boot:spring-boot-starter-aop:jar:2.1.2.RELEASE:compile
[INFO] | | +- org.springframework:spring-aop:jar:5.1.4.RELEASE:compile
[INFO] | | \- org.aspectj:aspectjweaver:jar:1.9.2:compile
[INFO] | +- org.springframework.boot:spring-boot-starter-jdbc:jar:2.1.2.RELEASE:compile
[INFO] | | +- com.zaxxer:HikariCP:jar:3.2.0:compile
[INFO] | | \- org.springframework:spring-jdbc:jar:5.1.4.RELEASE:compile
[INFO] | +- javax.transaction:javax.transaction-api:jar:1.3:compile
[INFO] | +- javax.xml.bind:jaxb-api:jar:2.3.1:compile
[INFO] | | \- javax.activation:javax.activation-api:jar:1.2.0:compile
[INFO] | +- org.hibernate:hibernate-core:jar:5.3.7.Final:compile
[INFO] | | +- org.jboss.logging:jboss-logging:jar:3.3.2.Final:compile
[INFO] | | +- javax.persistence:javax.persistence-api:jar:2.2:compile
[INFO] | | +- org.javassist:javassist:jar:3.23.1-GA:compile
[INFO] | | +- net.bytebuddy:byte-buddy:jar:1.9.7:compile
[INFO] | | +- antlr:antlr:jar:2.7.7:compile
[INFO] | | +- org.jboss:jandex:jar:2.0.5.Final:compile
[INFO] | | +- com.fasterxml:classmate:jar:1.4.0:compile
[INFO] | | +- org.dom4j:dom4j:jar:2.1.1:compile
[INFO] | | \- org.hibernate.common:hibernate-commons-annotations:jar:5.0.4.Final:compile
[INFO] | +- org.springframework.data:spring-data-jpa:jar:2.1.4.RELEASE:compile
[INFO] | | +- org.springframework.data:spring-data-commons:jar:2.1.4.RELEASE:compile
[INFO] | | +- org.springframework:spring-orm:jar:5.1.4.RELEASE:compile
[INFO] | | +- org.springframework:spring-context:jar:5.1.4.RELEASE:compile
[INFO] | | | \- org.springframework:spring-expression:jar:5.1.4.RELEASE:compile
[INFO] | | +- org.springframework:spring-tx:jar:5.1.4.RELEASE:compile
[INFO] | | +- org.springframework:spring-beans:jar:5.1.4.RELEASE:compile
[INFO] | | \- org.slf4j:slf4j-api:jar:1.7.25:compile
[INFO] | \- org.springframework:spring-aspects:jar:5.1.4.RELEASE:compile
[INFO] +- com.h2database:h2:jar:1.4.197:compile
[INFO] \- org.springframework.boot:spring-boot-starter-test:jar:2.1.2.RELEASE:test
[INFO]   +- org.springframework.boot:spring-boot-starter:jar:2.1.2.RELEASE:compile
[INFO]   | +- org.springframework.boot:spring-boot:jar:2.1.2.RELEASE:compile
[INFO]   | +- org.springframework.boot:spring-boot-autoconfigure:jar:2.1.2.RELEASE:compile
[INFO]   | +- org.springframework.boot:spring-boot-starter-logging:jar:2.1.2.RELEASE:compile
[INFO]   | | +- ch.qos.logback:logback-classic:jar:1.2.3:compile
[INFO]   | | | \- ch.qos.logback:logback-core:jar:1.2.3:compile
[INFO]   | | +- org.apache.logging.log4j:log4j-to-slf4j:jar:2.11.1:compile
[INFO]   | | | \- org.apache.logging.log4j:log4j-api:jar:2.11.1:compile
[INFO]   | | \- org.slf4j:jul-to-slf4j:jar:1.7.25:compile
[INFO]   | +- javax.annotation:javax.annotation-api:jar:1.3.2:compile
[INFO]   | \- org.yaml:snakeyaml:jar:1.23:runtime
[INFO]   +- org.springframework.boot:spring-boot-test:jar:2.1.2.RELEASE:test
[INFO]   +- org.springframework.boot:spring-boot-test-autoconfigure:jar:2.1.2.RELEASE:test
[INFO]   +- com.jayway.jsonpath:json-path:jar:2.4.0:test
[INFO]   | \- net.minidev:json-smart:jar:2.3:test
[INFO]   |   \- net.minidev:accessors-smart:jar:1.2:test
[INFO]   |     \- org.ow2.asm:asm:jar:5.0.4:test
[INFO]   +- junit:junit:jar:4.12:test
[INFO]   +- org.assertj:assertj-core:jar:3.11.1:test
[INFO]   +- org.mockito:mockito-core:jar:2.23.4:test
[INFO]   | +- net.bytebuddy:byte-buddy-agent:jar:1.9.7:test
[INFO]   | \- org.objenesis:objenesis:jar:2.6:test
[INFO]   +- org.hamcrest:hamcrest-core:jar:1.3:test
[INFO]   +- org.hamcrest:hamcrest-library:jar:1.3:test
[INFO]   +- org.skyscreamer:jsonassert:jar:1.5.0:test
[INFO]   | \- com.vaadin.external.google:android-json:jar:0.0.20131108.vaadin1:test
[INFO]   +- org.springframework:spring-core:jar:5.1.4.RELEASE:compile
[INFO]   | \- org.springframework:spring-jcl:jar:5.1.4.RELEASE:compile
[INFO]   +- org.springframework:spring-test:jar:5.1.4.RELEASE:test
[INFO]   \- org.xmlunit:xmlunit-core:jar:2.6.2:test
```

3. Spring Data JPA

Book.java

```
package com.mkyong;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Book {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String name;

    public Book() {
    }

    public Book(String name) {
        this.name = name;
    }

    //getters, setters, toString...
}
```

BookRepository.java

```
package com.mkyong;

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface BookRepository extends CrudRepository<Book, Long> {

    List<Book> findByName(String name);

}
```

4. @SpringBootApplication

4.1 Start Spring Boot, inserts 3 books into the H2 database, and test the `find()` method.

StartApplication.java

```
package com.mkyong;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class StartApplication implements CommandLineRunner {

    private static final Logger log = LoggerFactory.getLogger(StartApplication.class);

    @Autowired
    private BookRepository repository;

    public static void main(String[] args) {
        SpringApplication.run(StartApplication.class, args);
    }

    @Override
    public void run(String... args) {

        log.info("StartApplication...");

        repository.save(new Book("Java"));
        repository.save(new Book("Node"));
        repository.save(new Book("Python"));

        System.out.println("\nfindAll()");
        repository.findAll().forEach(x -> System.out.println(x));

        System.out.println("\nfindById(1L)");
        repository.findById(1L).ifPresent(x -> System.out.println(x));

        System.out.println("\nfindByName('Node')");
        repository.findByName("Node").forEach(x -> System.out.println(x));

    }

}
```

5. HikariCP Connection Pool

By default, Spring Boot 2 uses HikariCP as the default connection pool.

5.1 Enable the logging level of `com.zaxxer` to debug, it will print out the default HikariCP configuration.

application.properties
logging.level.org.springframework=INFO logging.level.com.mkyong=INFO logging.level.com.zaxxer=DEBUG logging.level.root=ERROR

5.2 Run the Spring Boot, review the logging :

```
DEBUG com.zaxxer.hikari.HikariConfig - Driver class org.h2.Driver found in Thread context class loader jdk.interr
DEBUG com.zaxxer.hikari.HikariConfig - HikariPool-1 - configuration:
DEBUG com.zaxxer.hikari.HikariConfig - allowPoolSuspension.....false
DEBUG com.zaxxer.hikari.HikariConfig - autoCommit.....true
DEBUG com.zaxxer.hikari.HikariConfig - catalog.....none
DEBUG com.zaxxer.hikari.HikariConfig - connectionInitSql.....none
DEBUG com.zaxxer.hikari.HikariConfig - connectionTestQuery.....none
DEBUG com.zaxxer.hikari.HikariConfig - connectionTimeout.....30000
DEBUG com.zaxxer.hikari.HikariConfig - dataSource.....none
DEBUG com.zaxxer.hikari.HikariConfig - dataSourceClassName.....none
DEBUG com.zaxxer.hikari.HikariConfig - dataSourceJNDI.....none
DEBUG com.zaxxer.hikari.HikariConfig - dataSourceProperties.....{password=<masked>}
DEBUG com.zaxxer.hikari.HikariConfig - driverClassName....."org.h2.Driver"
DEBUG com.zaxxer.hikari.HikariConfig - healthCheckProperties.....{}
DEBUG com.zaxxer.hikari.HikariConfig - healthCheckRegistry.....none
DEBUG com.zaxxer.hikari.HikariConfig - idleTimeout.....600000
DEBUG com.zaxxer.hikari.HikariConfig - initializationFailTimeout.....1
DEBUG com.zaxxer.hikari.HikariConfig - isolateInternalQueries.....false
DEBUG com.zaxxer.hikari.HikariConfig - jdbcUrl.....jdbc:h2:mem:testdb;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE
DEBUG com.zaxxer.hikari.HikariConfig - leakDetectionThreshold.....0
DEBUG com.zaxxer.hikari.HikariConfig - maxLifetime.....1800000
DEBUG com.zaxxer.hikari.HikariConfig - maximumPoolSize.....10
DEBUG com.zaxxer.hikari.HikariConfig - metricRegistry.....none
DEBUG com.zaxxer.hikari.HikariConfig - metricsTrackerFactory.....none
DEBUG com.zaxxer.hikari.HikariConfig - minimumIdle.....10
DEBUG com.zaxxer.hikari.HikariConfig - password.....<masked>
DEBUG com.zaxxer.hikari.HikariConfig - poolName....."HikariPool-1"
DEBUG com.zaxxer.hikari.HikariConfig - readOnly.....false
DEBUG com.zaxxer.hikari.HikariConfig - registerMbeans.....false
DEBUG com.zaxxer.hikari.HikariConfig - scheduledExecutor.....none
DEBUG com.zaxxer.hikari.HikariConfig - schema.....none
DEBUG com.zaxxer.hikari.HikariConfig - threadFactory.....internal
DEBUG com.zaxxer.hikari.HikariConfig - transactionIsolation.....default
DEBUG com.zaxxer.hikari.HikariConfig - username....."sa"
DEBUG com.zaxxer.hikari.HikariConfig - validationTimeout.....5000
INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Starting...
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:testdb user=SA
INFO com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Start completed.
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Pool stats (total=1, active=0, idle=1, waiting=0)
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn1: url=jdbc:h2:mem:testdb user=SA
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn2: url=jdbc:h2:mem:testdb user=SA
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn3: url=jdbc:h2:mem:testdb user=SA
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn4: url=jdbc:h2:mem:testdb user=SA
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn5: url=jdbc:h2:mem:testdb user=SA
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn6: url=jdbc:h2:mem:testdb user=SA
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn7: url=jdbc:h2:mem:testdb user=SA
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn8: url=jdbc:h2:mem:testdb user=SA
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - Added connection conn9: url=jdbc:h2:mem:testdb user=SA
DEBUG com.zaxxer.hikari.pool.HikariPool - HikariPool-1 - After adding stats (total=10, active=0, idle=10, waiting=0)
```

5.2 We can use `spring.datasource.hikari.*` to override the default settings.

```
application.properties

logging.level.org.springframework=INFO
logging.level.com.mkyong=INFO
logging.level.com.zaxxer=DEBUG
logging.level.root=ERROR

spring.datasource.hikari.connectionTimeout=20000
spring.datasource.hikari.maximumPoolSize=5
spring.datasource.hikari.poolName=HikariPoolZZZ
```

6. Unit Test

6.1 `@DataJpaTest` and `TestEntityManager` to test a Spring data JPA repository.

```
BookRepositoryTest.java
```

```
package com.mkyong;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.autoconfigure.orm.jpa.DataJpaTest;
import org.springframework.boot.test.autoconfigure.orm.jpa.TestEntityManager;
import org.springframework.test.context.junit4.SpringRunner;

import java.util.List;

import static org.assertj.core.api.Assertions.assertThat;
import static org.junit.Assert.assertEquals;

@RunWith(SpringRunner.class)
@DataJpaTest
public class BookRepositoryTest {

    @Autowired
    private TestEntityManager entityManager;

    @Autowired
    private BookRepository repository;

    @Test
    public void testFindByName() {

        entityManager.persist(new Book("C++"));

        List<Book> books = repository.findByName("C++");
        assertEquals(1, books.size());

        assertThat(books).extracting(Book::getName).containsOnly("C++");

    }

}
```

7. Demo

```
$ mvn spring-boot:run

INFO  com.mkyong.StartApplication - Started StartApplication in 2.79 seconds (JVM running for 10.883)
INFO  com.mkyong.StartApplication - StartApplication...

findAll()
DEBUG com.zaxxer.hikari.pool.PoolBase - HikariPoolZZZ - Reset (readOnly) on connection conn0: url=jdbc:h2:mem:tes
Book{id=1, name='Java'}
Book{id=2, name='Node'}
Book{id=3, name='Python'}

findById(1L)
DEBUG com.zaxxer.hikari.pool.PoolBase - HikariPoolZZZ - Reset (readOnly) on connection conn0: url=jdbc:h2:mem:tes
Book{id=1, name='Java'}

findByName('Node')
Book{id=2, name='Node'}
```

Download Source Code

```
$ git clone https://github.com/mkyong/spring-boot.git
$ cd spring-data-jpa
$ mvn spring-boot:run
```

References

- [HikariCP](#)
- [Spring Data JPA docs](#)
- [Working with SQL Databases](#)
- [Common application properties](#)
- [JUnit – How to test a List](#)
- [DataJpaTest docs](#)
- [TestEntityManager docs](#)
- [Spring Boot + Spring Data JPA + Oracle example](#)
- [Spring Boot Accessing Data with JPA](#)

ARE YOU A TINDER EXPERT? SWIPE FOR SUCCESS HERE

How many times does the average Tinder user log on per day?

15

▼

A 26

B 11

C 6

D 19

Your Score 0 Question 1/10

Xiaomi Redmi Note 7 6GB/...

4.799.000₫

Xiaomi Redmi Note 7 4GB/...

4.990.000₫

Xiaomi Redmi Note 7 4GB/...

4.990.000₫

About the Author



mkyong
Founder of [Mkyong.com](#), love Java and open source stuff. Follow him on [Twitter](#), or befriend him on [Facebook](#) or [Google Plus](#). If you like my tutorials, consider make a donation to [these charities](#).

Comments



Start the discussion...

Favorites Links

- [Android Getting Started](#)
- [Google App Engine – Java](#)
- [Spring 2.5.x Documentation](#)
- [Spring 3.2.x Documentation](#)
- [Spring 4.1.x Documentation](#)
- [Java EE 5 Tutorial](#)
- [Java EE 6 Tutorial](#)
- [Java EE 7 Tutorial](#)
- [Java 6 API](#)
- [Java 7 API](#)
- [Java 8 API](#)
- [Oracle J2SE Tutorials](#)
- [JSF Home Page](#)
- [JSP Home Page](#)
- [Maven Central Repository](#)

Partners & Bookmarks

- [JAX London](#)
- [Java Code Geeks](#)

About Mkyong.com

Mkyong.com is for Java and J2EE developers, all examples are simple and easy to understand, and well tested in our development environment.

Mkyong.com is created, written by, and maintained by Yong Mook Kim, aka Mkyong.

- [Gradle User Guide](#)
- [Hibernate ORM](#)
- [JAX-WS Home Page](#)
- [JAX-RS Home Page \(Jersey\)](#)
- [Tomcat 8 Documentation](#)

Copyright © 2008-2019 Mkyong.com, all rights reserved. [Privacy Policy](#)