



eritec

# Java & JDBC for Oracle DBAs

- and then some more

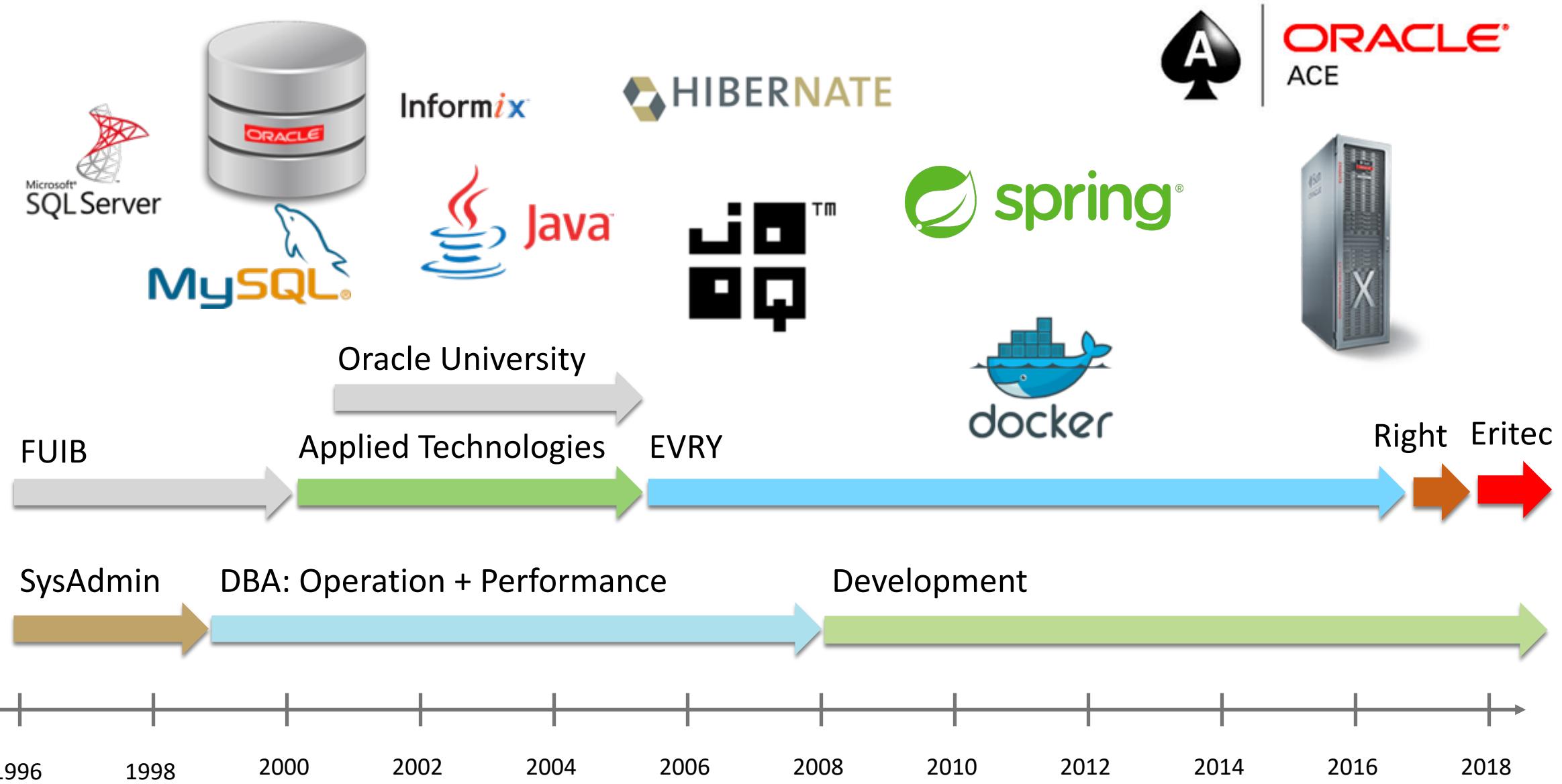
Lasse Jenssen



Lasse Jenssen

[www.jcon.no/oracle](http://www.jcon.no/oracle)

@lasjen [lasse.jenssen@eritec.no](mailto:lasse.jenssen@eritec.no)





**Why would an Oracle DBA go to a Java conference?**

# Why ?

Testing ...

JDBC features

Instrumentation

ORM APIs

Application Continuity

Calling PLSQL from Java

Communicating with Developers

Reading Code / Code Review

Writing production code

Or just for fun!

## *Warning!*

This is not a programming class, but an introduction to how you can get started writing Java code towards an Oracle database.

# Agenda



1

Install & Setup



2

Key knowledge



3

The "real" knowledge

# Agenda



1

Install & Setup



2

Key knowledge



3

The "real" knowledge

# Install & setup



1

Install & Setup

<http://www.json.no/oracle/?p=1668>

- [Part 1: Install/setup Oracle database \(in docker\)](#)
- [Part 2: Installing Java \(JDK\), Eclipse and Maven](#)
- [Part 3: Git, Oracle schemas and your first Java application](#)
- [Part 4: Your first JDBC Application](#)
- [Part 5: Spring-Boot, JdbcTemplate & DB migration \(Using FlywayDB\)](#)
- [Part 6: Spring-boot, JPA and Hibernate \(this\)](#)

# The “real” knowledge



3

The "real" knowledge

<https://github.com/lasjen/JavaForOracleDBAs>

- [Part 1: Install/setup Oracle database \(in docker\)](#)
- [Part 2: Installing Java \(JDK\), Eclipse and Maven](#)
- [Part 3: Git, Oracle schemas and your first Java application](#)
- [Part 4: Your first JDBC Application](#)
- [Part 5: Spring-Boot, JdbcTemplate & DB migration \(Using FlywayDB\)](#)
- [Part 6: Spring-boot, JPA and Hibernate \(this\)](#)

# This presentation



2

Key knowledge



Search or jump to...



Pull requests Issues Marketplace Explore



+ ▾

lasjen / OGYatraJavaForDBAs

Unwatch 1

Star 0

Fork

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Branch: master ▾

Go to file

Add file ▾

Code ▾

## About

Example code from Oracle Groundbreakers Yatra 2020 conference

Readme

## Releases

No releases published  
[Create a new release](#)

## Packages

No packages published  
[Publish your first package](#)

## Languages

PLSQL 50.6% Java 49.4%

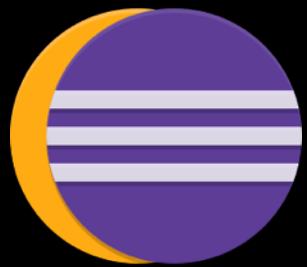
lasjen committed 47729e3 2 minutes ago <a href="#">...</a>	3 commits  1 branch  0 tags
FlywayDemo	Some new projects
HelloJdbc	Some new projects
HelloWorld	First update
HelloWorldMaven	First update
SpringBootDemo	Some new projects
README.md	Initial commit

README.md



# OGYatraJavaForDBAs

Example code from Oracle Groundbreakers Yatra 2020 conference





```
$ docker ps -a
```

CONTAINER ID	IMAGE	STATUS	PORTS	NAMES
632f9c634f2e	oracle/database:12.1.0.2-ee	Up 2 days (healthy)	0.0.0.0:1523->1521/tcp	ora12c-db-3
9cf2c97821e1	oracle/database:19.3.0-ee	Exited (255) 3 weeks ago	0.0.0.0:1522->1521/tcp	ora19-db-02
13187c2f0455	oracle/database:12.2.0.1-ee	Exited (255) 2 days ago	0.0.0.0:1521->1521/tcp	ora12c-db-1

## Java Platform, Standard Edition

### Java SE 11.0.1(LTS)

Java SE 11.0.1 is the latest release for the Java SE 11 Platforms  
[Learn more →](#)

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle JDK License](#)
- [Java SE Licensing Information User Manual](#)
  - Includes Third Party Licenses
- [Certified System Configurations](#)
- [Readme](#)

#### Looking for Oracle OpenJDK builds?

- **Oracle Customers and ISVs targeting Oracle LTS releases:** Oracle JDK is Oracle's supported Java SE version for customers and for developing, testing, prototyping or demonstrating your Java applications.
- **End users and developers looking for free JDK versions:** Oracle OpenJDK offers the same features and performance as Oracle JDK under the [GPL license](#).

**Oracle JDK**  
[DOWNLOAD](#) 



The screenshot shows the Oracle Java Downloads page. At the top is the Java logo. Below it is a navigation bar with links for Apps, Docker, Oracle, Privat, TOREAD, Guitar, Java, Oracle Blogs, and Right. The main content area has a sidebar with links for Java SE, Java EE, and Java ME. The main content area contains a large "Java SE Downloads" button and other navigation links for Overview, Downloads, Documentation, and Community.

# OpenJDK

## *Different Java files:*

File types	Example	Description
*.java	HelloWorld.java	Java Source Code
*.class	HelloWorld.class	Compiled Java Bytecode, which can be ran in a JVM
*.jar	HelloApp.jar	Java Archive. A zipped file containing the compressed versions of .class files and resources of compiled Java libraries and applications. We can run a Jar file from the command file (if built as an executable JAR).
*.war	WebHelloApp.war	Web Application Archive. Deploy on any Servlet/JSP container.

> javac HelloWorld.java

=> HelloWorld.class

> jar -cvf Hello.jar HelloWorld.class

=> Hello.jar

```
$ cat HelloWorld.java
class HelloWorld {
    public static void main(String args[]) {
        System.out.println("Hello, OGyatras 2020!");
    }
}
```

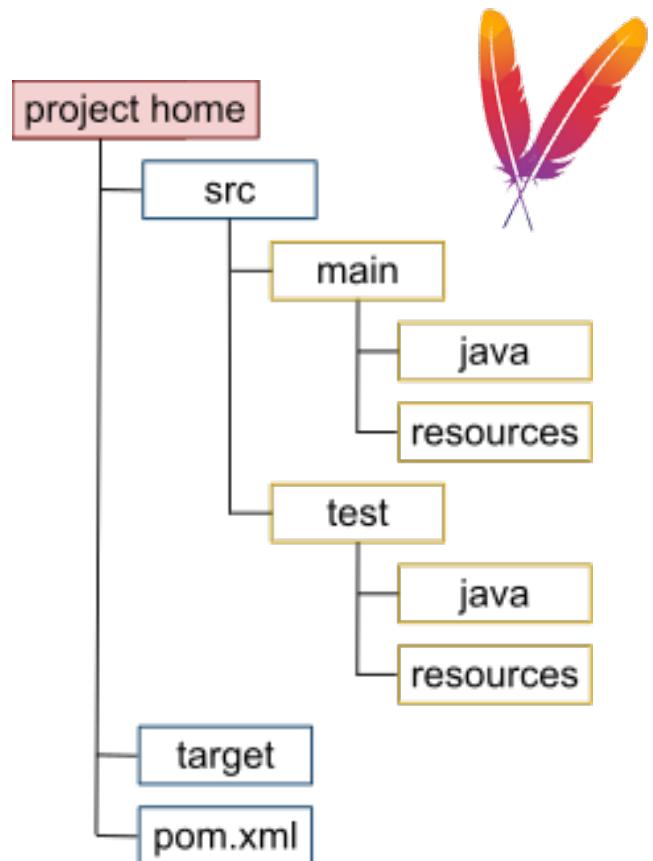
```
$ javac HelloWorld.java
$ ls
HelloWorld.class      HelloWorld.java
```

```
$ java HelloWorld
Hello, World
```

```
$ java -jar Hello.jar
Hello, OGyatras 2020!
```

```
$ cat Hello.mf
Manifest-Version: 1.0
Main-Class: HelloWorld
```

```
$ jar cmf Hello.mf Hello.jar HelloWorld.class
$ ls
Hello.jar          Hello.mf          HelloWorld.class      HelloWorld.java
```



## Maven

A standard way to **build** the projects, a clear definition of what the project consisted of (**POM**), an easy way to publish project information and a way to share JARs across several projects.

- Dependency management

[Welcome](#)[License](#)[ABOUT MAVEN](#)[What is Maven?](#)[Features](#)[Download](#)[Use](#)[Release Notes](#)[DOCUMENTATION](#)[Maven Plugins](#)[Index \(category\)](#)[User Centre](#)[Plugin Developer](#)[Centre](#)[Maven Central](#)[Repository](#)

# Downloading Apache Maven 3.6.3

Apache Maven 3.6.3 is the latest release and recommended version for all users.

The currently selected download mirror is <https://apache.uib.no/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, there are *backup* mirrors (at the end of the mirrors list) that should be available. You may also consult the [complete list of mirrors](#).

Other mirrors:

## System Requirements

**Java** Maven 3.3+ require JDK 1.7 or above to execute - they still allow you to build against 1.3 and other JDK versions by Using

**Development Toolchains**

**Kit (JDK)**

**Memory** No minimum requirement

**Disk** Approximately 10MB is required for the Maven installation itself. In addition to that, additional disk space will be used for your local Maven repository. The size of your local repository will vary depending on usage but expect at least 500MB.

[Welcome](#)[License](#)[ABOUT MAVEN](#)[What is Maven?](#)[Features](#)[Download](#)[Use](#)[Install](#)[Run](#)[Configure](#)[IDE Integration](#)[Support and Training](#)[Release Notes](#)[DOCUMENTATION](#)[Maven Plugins](#)[Index \(category\)](#)[User Centre](#)[Plugin Developer Centre](#)[Maven Central](#)[Repository](#)

# Installing Apache Maven

The installation of Apache Maven is a simple process of extracting the archive and adding the `bin` folder with the `mvn` command to the `PATH`.

Detailed steps are:

- Ensure `JAVA_HOME` environment variable is set and points to your JDK installation
- Extract distribution archive in any directory

```
1. unzip apache-maven-3.6.3-bin.zip
```

or

```
1. tar xzvf apache-maven-3.6.3-bin.tar.gz
```

Alternatively use your preferred archive extraction tool.

- Add the `bin` directory of the created directory `apache-maven-3.6.3` to the `PATH` environment variable
- Confirm with `mvn -v` in a new shell. The result should look similar to

```
1. Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
2. Maven home: /opt/apache-maven-3.6.3
3. Java version: 1.8.0_45, vendor: Oracle Corporation
4. Java home: /Library/Java/JavaVirtualMachines/jdk1.8.0_45.jdk/Contents/Home/jre
5. Default locale: en_US, platform encoding: UTF-8
6. OS name: "mac os x", version: "10.8.5", arch: "x86_64", family: "mac"
```

```
$ cd eclipse-workspace

$ mvn --version
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T20:33:14+0
Maven home: /Users/lassejenssen/Applications/apache-maven-3.5.4
Java version: 1.8.0_231, vendor: Oracle Corporation, runtime: /Library/Java/JavaVir
Default locale: en_NO, platform encoding: UTF-8
OS name: "mac os x", version: "10.15.5", arch: "x86_64", family: "mac"

$ mvn archetype:generate -DgroupId=no.eritec.demo -DartifactId=HelloWorldMaven
                           -DarchetypeArtifactId=maven-archetype-quickstart
                           -DarchetypeVersion=1.4 -DinteractiveMode=false

$ ls
HelloWorld      HelloWorldMaven

$ cd HelloWorldMaven
$ ls -ltr
total 8
-rw-r--r--  1 lassejenssen  staff   2668 Jul  5 00:48 pom.xml
drwxr-xr-x  4 lassejenssen  staff    128 Jul  5 00:48 src
```

```
$ tree
```

```
.
```

```
└── pom.xml
```

```
└── src
```

```
    ├── main
```

```
    │   └── java
```

```
    │       └── no
```

```
    │           └── eritec
```

```
    │               └── demo
```

```
    │                   └── App.java
```

```
    └── test
```

```
        └── java
```

```
            └── no
```

```
                └── eritec
```

```
                    └── demo
```

```
                        └── AppTest.java
```

11 directories, 3 files

```
$ cat src/main/java/no/eritec/demo/App.java
```

```
package no.eritec.demo;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )
    {
        System.out.println( "Hello World!" );
    }
}
```

```
$ cat src/test/java/no/eritec/demo/AppTest.java
```

```
package no.eritec.demo;

import static org.junit.Assert.assertTrue;

import org.junit.Test;

/**
 * Unit test for simple App.
 */
public class AppTest
{
    /**
     * Rigorous Test :-)
     */
    @Test
    public void shouldAnswerWithTrue()
    {
        assertTrue( true );
    }
}
```

```
$ cat 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-
  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>

  <groupId>no.eritec.demo</groupId>
  <artifactId>HelloWorldMaven</artifactId>
  <version>1.0-SNAPSHOT</version>

  <name>HelloWorldMaven</name>
  <properties>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

</project>
```

```
$ mvn compile  
[INFO] Scanning for projects...  
[INFO]  
[INFO] -----< no.eritec.demo:HelloWorldMaven >-----  
[INFO] Building HelloWorldMaven 1.0-SNAPSHOT  
[INFO] ----- [ jar ] -----  
[INFO]  
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ HelloWorldMaven ---  
[INFO] Using 'UTF-8' encoding to copy filtered resources.  
[INFO]  
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ HelloWorldMaven ---  
[INFO] Changes detected - recompiling the module!  
[INFO] Compiling 1 source file to .../HelloWorldMaven/target/classes  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 1.110 s  
[INFO] Finished at: 2020-07-05T01:18:07+02:00  
[INFO] -----
```

```
$ ls -ltr
total 8
drwxr-xr-x  4 lassejenssen  staff   128 Jul  5 00:48 src
-rw-r--r--  1 lassejenssen  staff   848 Jul  5 01:13 pom.xml
drwxr-xr-x  5 lassejenssen  staff   160 Jul  5 01:18 target
```

```
$ tree target/
target/
├── classes
│   └── no
│       └── eritec
│           └── demo
│               └── App.class
├── generated-sources
│   └── annotations
└── maven-status
    └── maven-compiler-plugin
        └── compile
            └── default-compile
                ├── createdFiles.lst
                └── inputFiles.lst
```

10 directories, 3 files

```
$ mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----< no.eritec.demo:HelloWorldMaven >-----
[INFO] Building HelloWorldMaven 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ HelloWorldMaven ---
[INFO] Deleting /Users/lassejenssen/git-repos/OGYatraJavaForDBAs>HelloWorldMaven/target
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.415 s
[INFO] Finished at: 2020-07-05T01:23:07+02:00
[INFO] -----
```

```
$ ls -ltr
total 8
drwxr-xr-x  4 lassejenssen  staff   128 Jul  5 00:48 src
-rw-r--r--  1 lassejenssen  staff   848 Jul  5 01:13 pom.xml
```

```
$ mvn compile package
[INFO] Scanning for projects...
[INFO] -----< no.eritec.demo:HelloWorldMaven >-----
[INFO] Building HelloWorldMaven 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
...
-----
-----  
T E S T S  
-----
Running no.eritec.demo.AppTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.045 sec

Results :
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ HelloWorldMaven ---
[INFO] Building jar: ./OGYatra.../HelloWorldMaven/target/HelloWorldMaven-1.0-SNAPSHOT.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.812 s
[INFO] Finished at: 2020-07-05T11:23:55+02:00
[INFO] -----
```

```
$ cat 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-
  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>

  <groupId>no.eritec.demo</groupId>
  <artifactId>HelloWorldMaven</artifactId>
  <version>1.0-SNAPSHOT</version>

  <name>HelloWorldMaven</name>
  ...
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.11</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>...</plugins>
  <build>
</project>
```

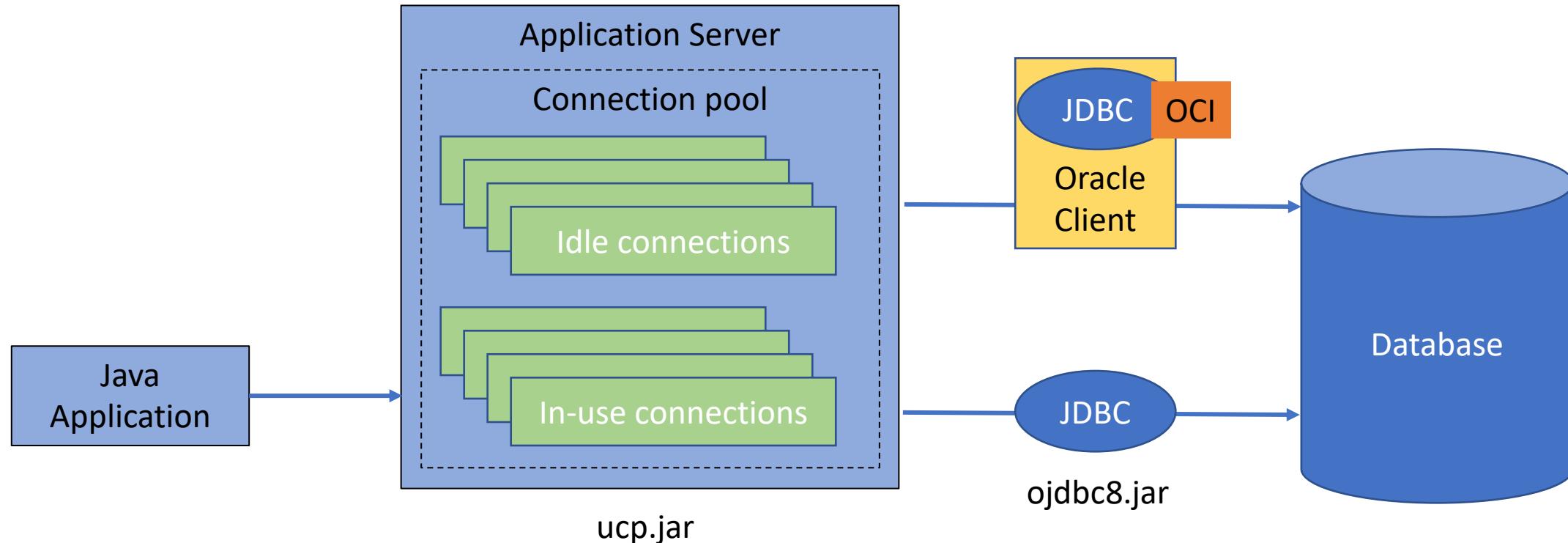
```
$ cat pom.xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>2.4</version>
      <configuration>
        <archive>
          <manifest>
            <mainClass>no.eritec.demo.App</mainClass>
          </manifest>
        </archive>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>
```

```
$ mvn clean compile package  
...  
[INFO]  
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ HelloWorldMaven ---  
[INFO] Building jar: ./OGYatra.../HelloWorldMaven/target/HelloWorldMaven-1.0-SNAPSHOT.jar  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 1.812 s  
[INFO] Finished at: 2020-07-05T11:23:55+02:00  
[INFO] -----
```

```
$ java -jar target/HelloWorldMaven-1.0-SNAPSHOT.jar  
Hello World!
```

JDBC OCI Driver (Type 4): Requires Client installation. Native C libraries



JDBC Thin Driver (Type 2): Pure Java using Java Socket (TCP/IP)

```
$ mvn archetype:generate -DgroupId=no.eritec.demo -DartifactId=HelloJdbc  
          -DarchetypeArtifactId=maven-archetype-quickstart  
          -DarchetypeVersion=1.4 -DinteractiveMode=false
```

```
$ vi 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/..."  
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/...">  
  <modelVersion>4.0.0</modelVersion>  
  
  <groupId>no.eritec.demo</groupId>  
  <artifactId>HelloWorldMaven</artifactId>  
  <version>1.0-SNAPSHOT</version>  
  ...  
  <dependencies>  
    ...  
  
    <dependency>  
      <groupId>com.oracle.jdbc</groupId>  
      <artifactId>ojdbc8</artifactId>  
      <version>12.2.0.1</version>  
    </dependency>  
  </dependencies>  
  
</project>
```

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

...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-jar-plugin</artifactId>
      <version>2.4</version>
      <configuration>
        <archive>
          <manifest>
            <addClasspath>true></addClasspath>
            <classpathPrefix>lib/</classpathPrefix>
            <mainClass>no.eritec.demo.App</mainClass>
          </manifest>
        </archive>
      </configuration>
    </plugin>
  </plugins>
<build>
</project>
```

...

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>2.5.1</version>
  <executions>
    <execution>
      <id>copy-dependencies</id>
      <phase>prepare-package</phase>
      <goals>
        <goal>copy-dependencies</goal>
      </goals>
      <configuration>
        <outputDirectory>${project.build.directory}/lib/</outputDirectory>
      </configuration>
    </execution>
  </executions>
</plugin>
</plugins>
<build>
</project>
```

```
package no.eritec.demo;

import java.sql.Connection;
...

public class App {
    public final static String DB_URL = "jdbc:oracle:thin:@localhost:1523/orcl";
    public final static String DB_USERNAME = "appdata";
    public final static String DB_PASSWORD = "app";

    public static void main(String[] args) throws SQLException{
        DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
        Connection conn = DriverManager.getConnection(DB_URL, DB_USERNAME, DB_PASSWORD);

        PreparedStatement ps = conn.prepareStatement("select user from dual");
        ResultSet rs = ps.executeQuery();

        while (rs.next()) {
            System.out.println("Result: " + rs.getString(1));
        }

        rs.close(); ps.close(); conn.close();
        System.out.println("Hello World!");
    }
}
```

```
$ java -jar target>HelloJdbc-1.0-SNAPSHOT.jar
```

```
Result: APPDATA
```

```
Hello World!
```



## Database Migration tool

- Command line
- Java API
- Maven Plugin
- Commands:
  - clean, migrate, info, validate, baseline, repair, undo

```
$ mvn archetype:generate -DgroupId=no.eritec.demo -DartifactId=HelloJdbc  
          -DarchetypeArtifactId=maven-archetype-quickstart  
          -DarchetypeVersion=1.4 -DinteractiveMode=false
```

```
$ vi 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/..."  
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/...">  
  <modelVersion>4.0.0</modelVersion>  
  
  <groupId>no.eritec.demo</groupId>  
  <artifactId>HelloWorldMaven</artifactId>  
  <version>1.0-SNAPSHOT</version>  
  ...  
  <dependencies>  
    ...  
  
    <dependency>  
      <groupId>org.flywaydb</groupId>  
      <artifactId>flyway-core</artifactId>  
      <version>6.5.0</version>  
    </dependency>  
  </dependencies>  
  
</project>
```

...

```
<build>
  <plugins>
    ...
    <plugin>
      <groupId>org.flywaydb</groupId>
      <artifactId>flyway-maven-plugin</artifactId>
      <configuration>
        <url>jdbc:oracle:thin:@localhost:1523/ORCL</url>
        <user>appdata</user>
        <password>app</password>
        <locations>db/migration/</locations>
        <table>SCHEMA_VERSION</table>
      </configuration>
      <dependencies>
        <dependency>
          <groupId>com.oracle.jdbc</groupId>
          <artifactId>ojdbc8</artifactId>
          <version>12.2.0.1</version>
        </dependency>
      </dependencies>
    </plugin>
  </plugins>
<build>
</project>
```

```
$ mvn archetype:generate -DgroupId=no.eritec.demo -DartifactId=FlywayDemo
                           -DarchetypeArtifactId=maven-archetype-quickstart
                           -DarchetypeVersion=1.4 -DinteractiveMode=false

$ mkdir -p FlywayDemo/src/main/resources/db/migration

$ cd FlywayDemo/src/main/resources/db/migration

$ vi V1_1__tbl_main.sql
$ vi V1_2__tbl_big_table.sql
$ vi R_3_1_1_type_main.sql
$ vi R_5_pkss_librarian.sql
$ vi R_6_pkss_librarian.sql
```

```
-- Table: AUTHORS (AUTS)                                     v1_1__tbl_main.sql
-----
create table authors (
    id          number(10),
    name        varchar2(100 char)
);

alter table authors add constraint auts_pk primary key (id);

-----
-- Table: BOOKS (BOOK)
-----

create sequence books_seq start with 1 increment by 1 cache 20;
create table books (
    id                  number(10)      default books_seq.nextval, -- PK
    title               varchar2(200 char) constraint book_title_nn not null,
    isbn                varchar2(20 char) constraint book_isbn_nn not null,
    year_of_publication number(4)       constraint book_yearofpubl_nn not null,
    fk_auths_id         number(10)
);

alter table books add (
    constraint book_pk primary key (id),
    constraint book_auth_fk foreign key (fk_auths_id) references authors (id)
);
```

-- Table: BIG\_TABLE

**v1\_2\_tbl\_big\_table.sql**

CREATE SEQUENCE big\_table\_seq START WITH 1 INCREMENT BY 1 CACHE 100;

CREATE TABLE BIG\_TABLE (

id	NUMBER(18)	default big_table_seq.nextval,
owner	VARCHAR2(128 CHAR)	NOT NULL,
object_name	VARCHAR2(128 CHAR)	NOT NULL,
subobject_name	VARCHAR2(128 CHAR),	
object_id	NUMBER	NOT NULL,
data_object_id	NUMBER,	
object_type	VARCHAR2(23 CHAR),	
created	TIMESTAMP (6),	
last_ddl_time	TIMESTAMP (6),	
timestamp	VARCHAR2(19 CHAR),	
status	VARCHAR2(7 CHAR),	
temporary	VARCHAR2(1 CHAR),	
generated	VARCHAR2(1 BYTE),	
secondary	VARCHAR2(1 CHAR),	
	constraint big_table_pk primary key (id)	

) ;

-- Types: BOOK\_OT & BOOKS\_CT

**R\_3\_1\_1\_type\_main.sql**

```
create or replace type BOOK_OT as object(
    title          varchar2(200 char),
    author         varchar2(100 char),
    isbn           varchar2(20 char),
    year_of_publication number(4)
)
/
```

```
create type BOOKS_CT is table of BOOK_OT
/
```

-- Package Specification: LIBRARIAN

**R\_\_5\_pks\_librarian.sql**

create or replace package librarian as

procedure insert\_books(books\_i in books\_ct);

function get\_books\_by\_author (author\_i in varchar2) return books\_ct;

end librarian;

/

```
create or replace package body LIBRARIAN as
```

## R\_6\_pkb\_librarian.sql

```
-- Procedure: INSERT_BOOKS
```

```
procedure insert_books(books_i in books_ct) is
```

```
begin
```

```
    merge into books b
```

```
        using ( select a.id atr_id, b.*
```

```
                  from authors a right join table( books_i ) b on (a.name = b.author)
                  ) newbook
```

```
    on ( b.isbn = newbook.isbn or
```

```
        (b.title = newbook.title and b.fk_arts_id = newbook.atr_id)
```

```
    )
```

```
    when matched
```

```
        then update set b.year_of_publication = newbook.year_of_publication
```

```
    when not matched
```

```
        then insert ( b.id, b.title, b.isbn, b.year_of_publication, b.fk_arts_id)
```

```
            values ( books_seq.nextval
```

```
                  , newbook.title, newbook.isbn, newbook.year_of_publication, newbook.atr_id
                  );
```

```
end insert_books;
```

```
...
```

```
end librarian;
```

```
/
```

```
$ cd ../../../../
```

```
FlywayDemo$ pwd
```

```
$ mvn clean compile flyway:info
```

```
[INFO] Flyway Community Edition 5.1.4 by Boxfuse
```

```
[INFO] Database: jdbc:oracle:thin:@localhost:1523/orcl (Oracle 12.1)
```

```
[INFO] Schema version: << Empty Schema >>
```

```
[INFO]
```

Category	Version	Description	Type	Installed On	State
Versioned	1.1	tbl main	SQL		Pending
Versioned	1.2	tbl big table	SQL		Pending
Repeatable		3 1 1 type main	SQL		Pending
Repeatable		5 pks librarian	SQL		Pending
Repeatable		6 pkb librarian	SQL		Pending

```
[INFO] -----
```

```
[INFO] BUILD SUCCESS
```

```
[INFO] -----
```

```
[INFO] Total time: 1.854 s
```

```
[INFO] Finished at: 2020-07-06T10:40:05+02:00
```

```
[INFO] -----
```

```
$ mvn flyway:migrate
[INFO] Scanning for projects...
[INFO] -----< no.eritec.demo:FlywayDemo >-----
[INFO] Building FlywayDemo 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO] --- flyway-maven-plugin:5.1.4:migrate (default-cli) @ FlywayDemo ---
[INFO] Flyway Community Edition 5.1.4 by Boxfuse
[INFO] Database: jdbc:oracle:thin:@localhost:1523/orcl (Oracle 12.1)
[INFO] Successfully validated 5 migrations (execution time 00:00.011s)
[INFO] Creating Schema History table: "APPDATA"."SCHEMA_VERSION"
[INFO] Current version of schema "APPDATA": << Empty Schema >>
[INFO] Migrating schema "APPDATA" to version 1.1 - tbl main
[INFO] Migrating schema "APPDATA" to version 1.2 - tbl big table
[INFO] Migrating schema "APPDATA" with repeatable migration 3 1 1 type main
[INFO] Migrating schema "APPDATA" with repeatable migration 5 pks librarian
[INFO] Migrating schema "APPDATA" with repeatable migration 6 pkb librarian
[INFO] Successfully applied 5 migrations to schema "APPDATA" (execution time 00:02.201s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.671 s
[INFO] Finished at: 2020-07-06T12:10:08+02:00
[INFO] -----
```

```
$ mvn flyway:info
[INFO]
[INFO] -----< no.eritec.demo:FlywayDemo >-----
[INFO] Building FlywayDemo 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- flyway-maven-plugin:5.1.4:info (default-cli) @ FlywayDemo ---
[INFO] Flyway Community Edition 5.1.4 by Boxfuse
[INFO] Database: jdbc:oracle:thin:@localhost:1523/orcl (Oracle 12.1)
[INFO] Schema version: 1.2
[INFO] +-----+-----+-----+-----+-----+
| Category | Version | Description | Type | Installed On | State |
+-----+-----+-----+-----+-----+
| Versioned | 1.1 | tbl main | SQL | 2020-07-06 12:59:41 | Success |
| Versioned | 1.2 | tbl big table | SQL | 2020-07-06 12:59:43 | Success |
| Repeatable | | 3 1 1 type main | SQL | 2020-07-06 12:59:43 | Success |
| Repeatable | | 5 pks librarian | SQL | 2020-07-06 12:59:43 | Success |
| Repeatable | | 6 pkb librarian | SQL | 2020-07-06 12:59:43 | Success |
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Finished at: 2020-07-06T12:59:49+02:00
[INFO] -----
```

Reports DBA

Connections

...ATA DOCKER\_03\_SYSTEM DOCKER\_01\_APPDATA~1 SCHEMA\_VERSION

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL Actions...

Docker01

DOCKER\_01\_APPDATA

Tables

- AUTHORS
- BIG\_TABLE
- BOOKS
- SCHEMA\_VERSION

Views

Indexes

Packages

Procedures

Functions

Operators

Queues

Queues Tables

Triggers

Types

Sequences

Materialized Views

Materialized View Logs

Synonyms

Public Synonyms

Database Links

Public Database Links

installed_rank	version	description	type	script	checksum
1	11.1	tbl main	SQL	V1_1__tbl_main.sql	85198
2	21.2	tbl big table	SQL	V1_2__tbl_big_table.sql	109542
3	(null)	3 1 1 type main	SQL	R_3_1_1_type_main.sql	-154342
4	(null)	5 pks librarian	SQL	R__5_pks_librarian.sql	-92367
5	(null)	6 pkb librarian	SQL	R__6_pkb_librarian.sql	100312

```
$ mvn flyway:clean
```

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----< no.eritec.demo:FlywayDemo >-----
[INFO] Building FlywayDemo 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- flyway-maven-plugin:5.1.4:clean (default-cli) @ FlywayDemo ---
[INFO] Flyway Community Edition 5.1.4 by Boxfuse
[INFO] Database: jdbc:oracle:thin:@localhost:1523/orcl (Oracle 12.1)
[INFO] Successfully cleaned schema "APPDATA" (execution time 00:00.939s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.294 s
[INFO] Finished at: 2020-07-06T12:14:11+02:00
[INFO] -----
```

Connections DOCKER\_01\_APPDATA DOCKER\_03\_SYSTEM DOCKER\_01\_APPDATA~1

Reports

DBA

Docker01

- DOCKER\_01\_APPDATA
  - Tables
  - Views
  - Indexes
  - Packages
  - Procedures
  - Functions
  - Operators
  - Queues
  - Queues Tables
  - Triggers
  - Types
  - Sequences
  - Materialized Views
  - Materialized View Logs
  - Synonyms
  - Public Synonyms
  - Database Links
  - Public Database Links
  - Directories
  - Editions
  - Application Express
  - Java

Worksheet Query Builder

1 `select * from tab;`

Query Result All Rows Fetched: 0 in 0.031 seconds

TNAME	TABTYPE	CLUSTERID
-------	---------	-----------



eclipse

[Members](#)[Working Groups](#)[Projects](#)[More](#) ▾[!\[\]\(9ad9446bc2b2cfb864c26975a60a8300\_img.jpg\) Download](#)[Home](#) / [Downloads](#) / [Packages](#) / [Release](#) / [Eclipse IDE Oxygen](#) / [3A](#)[Eclipse Installer](#)    [Eclipse Packages](#)

## Eclipse IDE Oxygen 3a Packages

### Eclipse IDE for Java EE Developers



334 MB 1,077,485 DOWNLOADS

Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn, EGit and others.

[Windows 32-bit 64-bit](#)[Mac Cocoa 64-bit](#)[Linux 32-bit 64-bit](#)

### Eclipse IDE for Java Developers



183 MB 532,890 DOWNLOADS

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

[Windows 32-bit 64-bit](#)[Mac Cocoa 64-bit](#)[Linux 32-bit 64-bit](#)

### Eclipse IDE for C/C++ Developers



204 MB 134,025 DOWNLOADS

An IDE for C/C++ developers with Mylyn integration.

[Windows 32-bit 64-bit](#)[Mac Cocoa 64-bit](#)[Linux 32-bit 64-bit](#)

Develop in the cloud  
using Eclipse Orion and  
deploy to IBM Cloud  
using Continuous Delivery

[» Learn more](#)

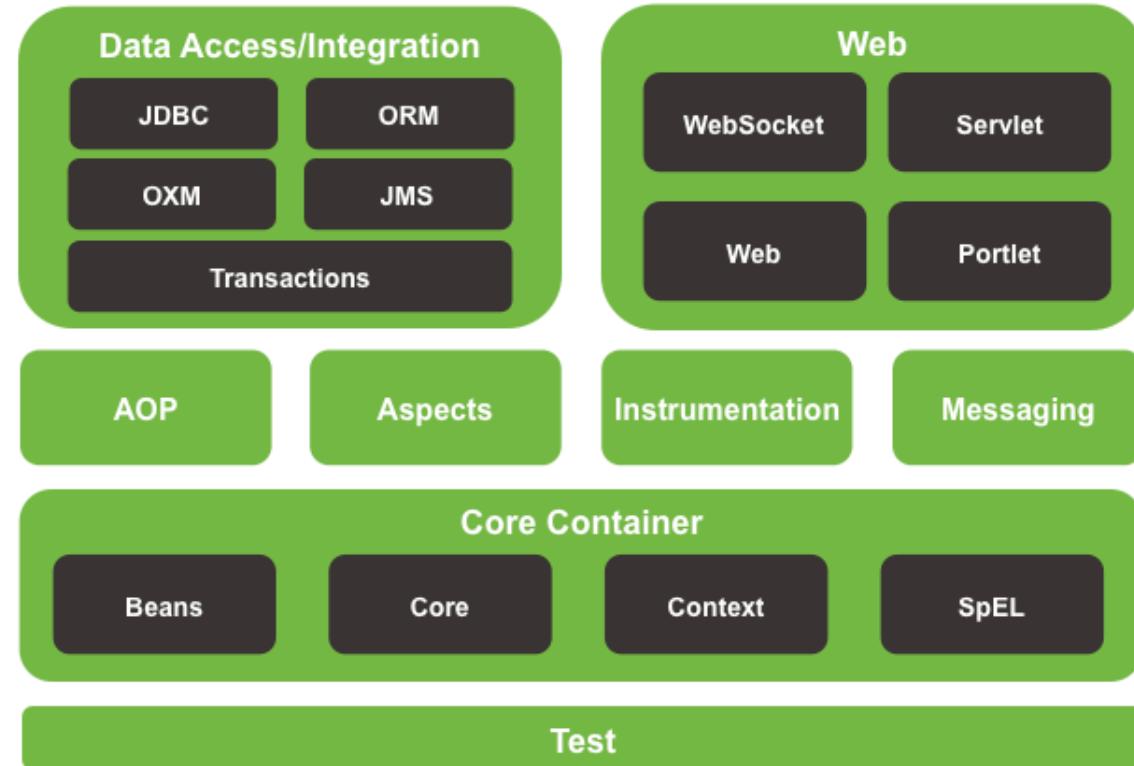
### Get Eclipse IDE 2020-06

Install your favorite desktop IDE  
packages.

[Download 64 bit](#)[Download Packages](#) | [Need Help?](#)

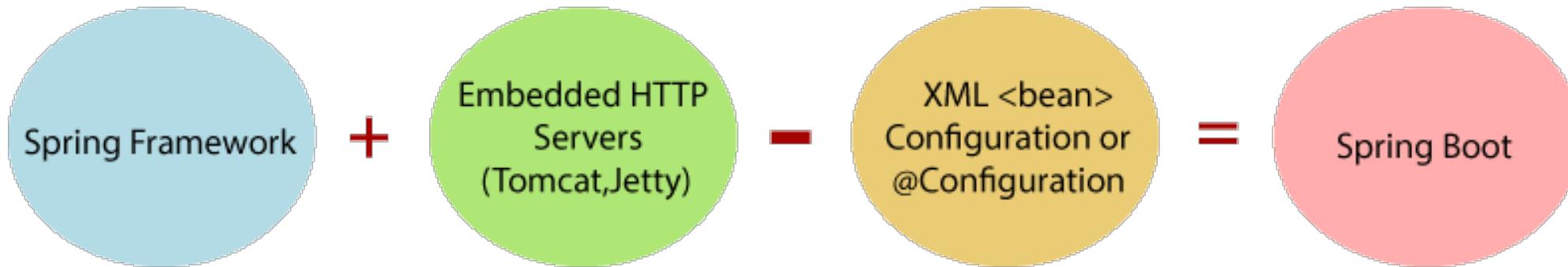


- Framework
- Dependency Injection / Inversion of Control
- Aspect oriented programming





- Rapid Application Development
- Avoid heavy XML config (Preconfigured)
- Starters: Collection of relevant dependencies





### Project

Maven Project

Gradle Project

### Language

Java

Kotlin

Groovy

### Dependencies

**ADD DEPENDENCIES... ⌘ + B**

No dependency selected

### Spring Boot

- 2.4.0 (SNAPSHOT)     2.4.0 (M1)     2.3.2 (SNAPSHOT)
- 2.3.1     2.2.9 (SNAPSHOT)     2.2.8     2.1.16 (SNAPSHOT)
- 2.1.15

### Project Metadata

Group no.eritec.demo

Artifact SpringBootDemo

Name SpringBootDemo

Description Demo project for Spring Boot



**GENERATE ⌘ + ↲**

**EXPLORE CTRL + SPACE**

**SHARE...**



# spring initializr

## Project

- Maven Project
- Gradle Project

## Language

- Java
- Kotlin
- Groovy

## Spring Boot

- 2.4.0 (SNAPSHOT)
- 2.4.0 (M1)
- 2.3.2 (SNAPSHOT)
- 2.3.1
- 2.2.9 (SNAPSHOT)
- 2.2.8
- 2.1.16 (SNAPSHOT)
- 2.1.15

## Project Metadata

Group no.eritec.demo

Artifact SpringBootDemo

Name SpringBootDemo

Description Demo project for Spring Boot

## Dependencies

[ADD DEPENDENCIES... ⌘ + B](#)

### Spring Data JPA

SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

### Oracle Driver

SQL

A JDBC driver that provides access to Oracle.

### Flyway Migration

SQL

Version control for your database so you can migrate from any version (incl. an empty database) to the latest version of the schema.

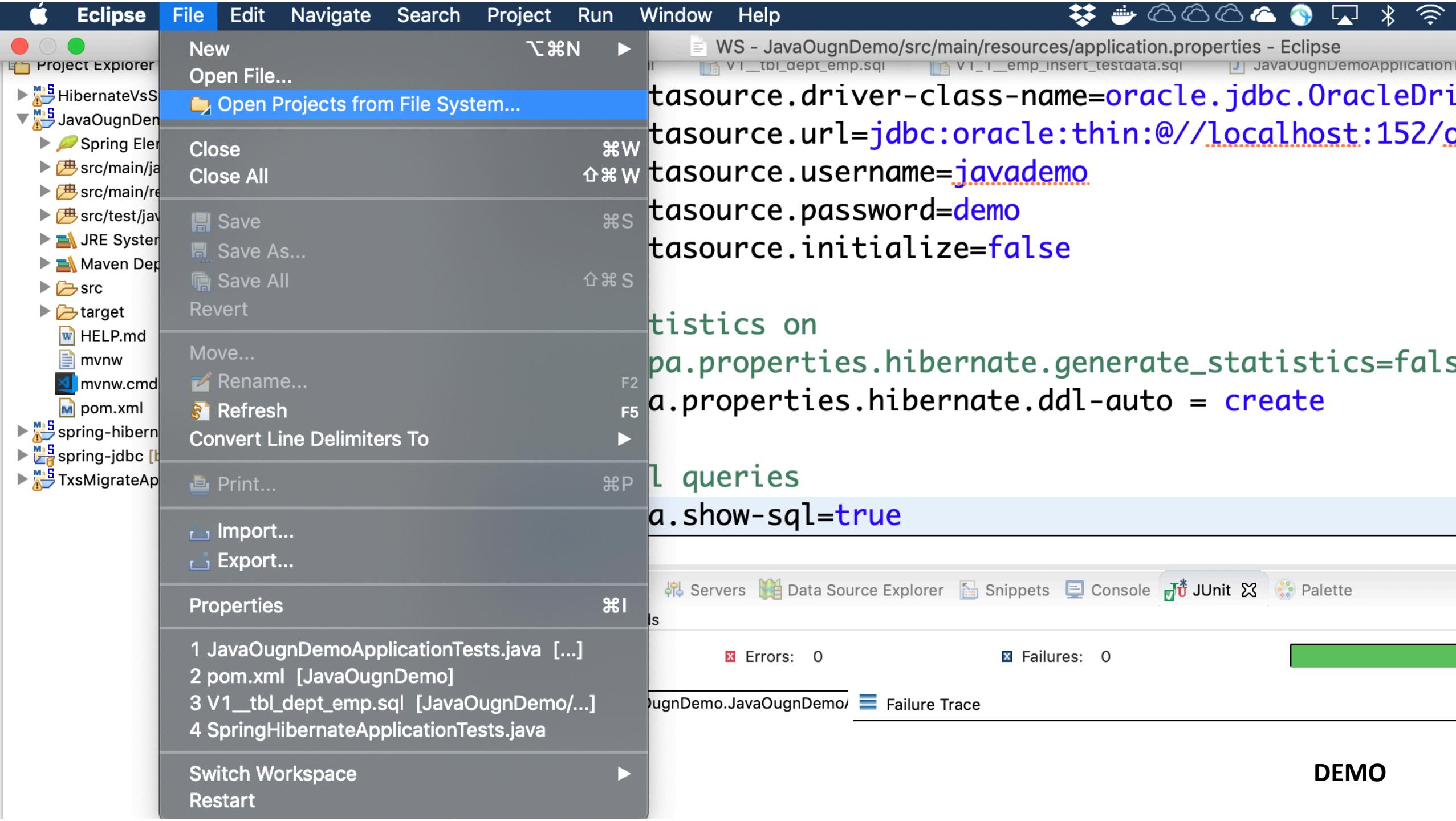


[GENERATE ⌘ + ↵](#)

[EXPLORE CTRL + SPACE](#)

[SHARE...](#)





## Entity class: Employee

```
package no.eritec.demo.SpringBootDemo.entity;

import java.util.Date;

public class Employee {

    private int empno;
    private String ename;
    private String job;
    private int managerNo;
    private Date hiredate;
    private int salary;
    private int comm;
    private int deptno;

    // Constructor
    public Employee() {
        super();
    }

    // Getters & Setters
    public Date getHiredate() {
        return hiredate;
    }

    public void setHiredate(Date hiredate) {
        this.hiredate = hiredate;
    }

    public int getEmpno() {
        return empno;
    }

    public void setEmpno(int empno) {
        this.empno = empno;
    }

    ...
}
```

## **DAO/Repository class – with JDBC: EmployeeJdbcRepository**

```
package no.eritec.demo.SpringBootDemo.repository;

import java.sql.Connection;
...
import javax.sql.DataSource;

import org.springframework.beans.factory.annotation.Autowired;

import no.eritec.demo.SpringBootDemo.entity.Employee;

@Repository
public class EmployeeJdbcRepository {
    @Autowired
    DataSource datasource;

    ...
}
```

## DAO/Repository class – with JDBC: EmployeeJdbcRepository

```
public List<Employee> findAll() throws SQLException {  
    Connection conn = datasource.getConnection();  
    Statement stmt = null;  
    String query = "select * from emp";  
    List<Employee> list = new ArrayList<Employee>();  
    try {  
        stmt = conn.createStatement();  
        ResultSet rs = stmt.executeQuery(query);  
        while (rs.next()) {  
            Employee emp = new Employee();  
            emp.setEmpno(rs.getInt("empno")); emp.setManagerNo(rs.getInt("mgr")); ...  
            list.add(emp);  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally {  
        if (stmt != null) { stmt.close(); }  
    }  
  
    return list;  
}
```

## DAO/Repository class – with Spring (JdbcTemplate) : EmployeeTemplateRepository

```
package no.eritec.demo.SpringBootDemo.repository;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
...
import no.eritec.demo.SpringBootDemo.entity.Employee;

@Repository
public class EmployeeTemplateRepository {
    @Autowired
    private JdbcTemplate jdbcTemplate;

    @Transactional(readOnly=true)
    public List<Employee> findAll() {
        return jdbcTemplate.query("select * from emp", new EmployeeRowMapper());
    }

    class EmployeeRowMapper implements RowMapper<Employee> { ...
    }
}
```

## **DAO/Repository class – with Spring (JdbcTemplate) :**

### **EmployeeTemplateRepository**

...

```
class EmployeeRowMapper implements RowMapper<Employee> {  
    @Override  
    public Employee mapRow(ResultSet rs, int rowNum) throws SQLException {  
        Employee e = new Employee();  
        e.setEmpno(rs.getInt("empno"));  
        e.setEname(rs.getString("ename"));  
        e.setJob(rs.getString("job"));  
        e.setManagerNo(rs.getInt("mgr"));  
        e.setHiredate(rs.getDate("hiredate"));  
        e.setSalary(rs.getInt("sal"));  
        e.setComm(rs.getInt("comm"));  
        e.setDeptno(rs.getInt("deptno"));  
  
        return e;  
    }  
}
```

## **DAO/Repository class – with Spring (JdbcTemplate) :**

### **EmployeeTemplateRepository**

```
@Transactional(readOnly=true)
public Employee findEmployeeById(int empno) {
    return jdbcTemplate.queryForObject("select * from emp where empno=?",
                                      new Object[]{empno}, new EmployeeRowMapper());
}

@Transactional(readOnly=true)
public List<Employee> findEmployeeByName(String ename) {
    return jdbcTemplate.query("select * from emp where ename like ?",
                           new Object[]{ename}, new EmployeeRowMapper());
}

@Transactional(readOnly=true)
public int deleteEmployeeById(int empno) {
    Object[] params = { empno };
    int[] types = {Types.BIGINT};

    return jdbcTemplate.update("delete from emp where empno=?", params, types);
}
```

## DAO/Repository class – with Spring (JdbcTemplate) : EmployeeTemplateRepository

```
public Employee create(final Employee emp) {  
    String id_column = "empno";  
    final String sql =  
        "insert into emp(ename,job,mgr,hiredate,sal,comm,deptno) values(?,?,?,?,?,?)";  
    KeyHolder holder = new GeneratedKeyHolder();  
    jdbcTemplate.update(conn -> {  
        PreparedStatement ps = conn.prepareStatement(sql, new String[]{id_column});  
        ps.setString(1, emp.getEname());  
        ps.setString(2, emp.getJob());  
        if (emp.getManagerNo() == 0) ps.setNull(3, Types.INTEGER);  
        else ps.setInt(3, emp.getManagerNo());  
        ps.setDate(4, new Date(emp.getHiredate().getTime()));  
        ps.setInt(5, emp.getSalary()); ps.setInt(6, emp.getComm());  
        ps.setInt(7, emp.getDeptno());  
        return ps;  
    }, holder);  
  
    int newEmpno = holder.getKey().intValue();  
    emp.setEmpno(newEmpno);  
    return emp;  
}
```

## Testing Repository classes: SpringBootDemoApplicationTests

```
package no.eritec.demo.SpringBootDemo;

import ...

@SpringBootTest
class SpringBootDemoApplicationTests {

    @Autowired
    EmployeeJdbcRepository empRepo;

    @Test
    void EmployeeJdbcRepositoryTest() throws SQLException {
        List<Employee> list = empRepo.findAll();

        assertEquals(3, list.size());
    }
}
```

What is *probably* the most hated word amongst  
DBAs working with Java developers?

HIBERNATE

“Your performance problems are caused by Hibernate”

vs

“Your performance problems are caused by you ...  
using Hibernate the wrong way!”

**Java Persistence API (JPA)**

Specification

**Object-Relational Mapping  
APIs (ORM APIs)**

Implementation  
*(for instance Hibernate)*

## Entity class: Department

```
package no.eritec.demo.SpringBootDemo.entity;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="dept")
public class Department {
    @Id
    private int deptno;
    private String dname;
    private String loc;

    public Department(int deptno, String dname, String loc) {
        this.deptno = deptno;
        this.dname = dname;
        this.loc = loc;
    }
}
```

## JPA Repository: DepartmentRepository

```
package no.eritec.demo.JavaOughDemo.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
import no.eritec.demo.JavaOughDemo.entity.Department;

@Repository
public interface DepartmentRepository extends JpaRepository<Department, Integer> {

}
```

## Testing Repository classes: **SpringBootDemoApplicationTests**

```
package no.eritec.demo.SpringBootDemo;

import ...

@SpringBootTest
class SpringBootDemoApplicationTests {

    @Autowired
    EmployeeJdbcRepository empRepo;

    @Autowired
    DepartmentRepository deptRepo;

    ...
}
```

## Testing Repository classes: `SpringBootDemoApplicationTests`

...

```
@SpringBootTest
class SpringBootDemoApplicationTests {
    ...
    @Test
    void DepartmentRepositoryTest() throws SQLException {
        List<Department> list = deptRepo.findAll();
        assertEquals(3, list.size());

        deptRepo.save(new Department(40, "DUMMY", "NOWHERE"));
        list = deptRepo.findAll();
        assertEquals(4, list.size());

        deptRepo.deleteById(40);
        assertEquals(3, deptRepo.count());
    }
}
```

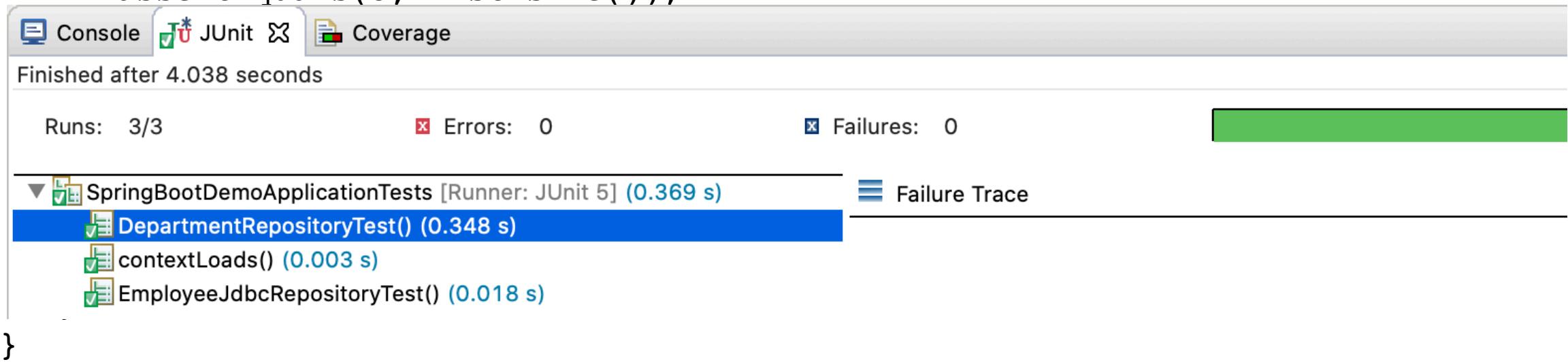
# Testing Repository classes: SpringBootDemoApplicationTests

...

```
@SpringBootTest
class SpringBootDemoApplicationTests {

    ...

    @Test
    void DepartmentRepositoryTest() throws SQLException {
        List<Department> list = deptRepo.findAll();
        assertEquals(3, list.size());
    }
}
```



# Testing Repository classes: SpringBootDemoApplicationTests

```
$ mvn clean compile package
...
2020-07-07 11:08:46.888  INFO 80073 --- [main] n.e.d.S.SpringBootDemoApplicationTests      : Started
SpringBootDemoApplicationTests in 3.321 seconds (JVM running for 4.117)
Hibernate: select department0_.deptno as deptno1_0_, department0_.dname as dname2_0_, department0_.loc
           as loc3_0_ from dept department0_
Hibernate: select department0_.deptno as deptno1_0_0_, department0_.dname as dname2_0_0_,
           department0_.loc as loc3_0_0_ from dept department0_ where department0_.deptno=?
Hibernate: insert into dept (dname, loc, deptno) values (?, ?, ?)
Hibernate: select department0_.deptno as deptno1_0_, department0_.dname as dname2_0_, department0_.loc
           as loc3_0_ from dept department0_
Hibernate: select department0_.deptno as deptno1_0_0_, department0_.dname as dname2_0_0_,
           department0_.loc as loc3_0_0_ from dept department0_ where department0_.deptno=?
Hibernate: delete from dept where deptno=?
Hibernate: select count(*) as col_0_0_ from dept department0_
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.104 s - in
no.eritec.demo.SpringBootDemo.SpringBootDemoApplicationTests
```

## Adding LOGGER: SpringBootDemoApplicationTests

```
@SpringBootTest
class SpringBootDemoApplicationTests {
    Logger log = LoggerFactory.getLogger(SpringBootDemoApplicationTests.class);

    ...
    @Test
    void DepartmentRepositoryTest() throws SQLException {
        List<Department> list = deptRepo.findAll();
        assertEquals(3, list.size());

        deptRepo.save(new Department(40, "DUMMY", "NOWHERE"));
        list = deptRepo.findAll();
        for (Department dept:list) {
            log.info("DEPARTMENT: " + dept.getDname());
        }
        assertEquals(4, list.size());

        deptRepo.deleteById(40);
        assertEquals(3, deptRepo.count());
    }
}
```

# Testing Repository classes: SpringBootDemoApplicationTests

```
$ mvn clean compile package
...
2020-07-07 11:43:41.215  INFO 80748 --- [           main] n.e.d.S.SpringBootDemoApplicationTests  :
    Started SpringBootDemoApplicationTests in 3.312 seconds (JVM running for 4.094)
Hibernate: select department0_.deptno as deptno1_0_, department0_.dname as dname2_0_, department0_.loc
    as loc3_0_ from dept department0_
Hibernate: select department0_.deptno as deptno1_0_0_, department0_.dname as dname2_0_0_,
    department0_.loc as loc3_0_0_ from dept department0_ where department0_.deptno=?
Hibernate: insert into dept (dname, loc, deptno) values (?, ?, ?)
Hibernate: select department0_.deptno as deptno1_0_, department0_.dname as dname2_0_, department0_.loc
    as loc3_0_ from dept department0_
2020-07-07 11:43:41.562  INFO 80748 --- [           main] n.e.d.S.SpringBootDemoApplicationTests  :
    DEPARTMENT: MANAGEMENT
2020-07-07 11:43:41.562  INFO 80748 --- [           main] n.e.d.S.SpringBootDemoApplicationTests  :
    DEPARTMENT: IT
2020-07-07 11:43:41.562  INFO 80748 --- [           main] n.e.d.S.SpringBootDemoApplicationTests  :
    DEPARTMENT: SALES
2020-07-07 11:43:41.562  INFO 80748 --- [           main] n.e.d.S.SpringBootDemoApplicationTests  :
    DEPARTMENT: DUMMY
Hibernate: select department0_.deptno as deptno1_0_0_, department0_.dname as dname2_0_0_,
    department0_.loc as loc3_0_0_ from dept department0_ where department0_.deptno=?
Hibernate: delete from dept where deptno=?
Hibernate: select count(*) as col_0_0_ from dept department0_
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.108 s - in
    no.eritec.demo.SpringBootDemo.SpringBootDemoApplicationTests
```

## EntityManager: EmployeeEMRepository

```
package no.eritec.demo.SpringJpaDemo.repository;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import org.springframework.stereotype.Repository;
import no.eritec.demo.SpringJpaDemo.entity.Employee;

@Repository
public class EmployeeEMRepository {
    @PersistenceContext
    EntityManager em;

    public Employee getEmployeeById(Integer empno) {
        Employee emp = (Employee) em.createQuery(
            "select e from Employee e join fetch e.department where e.empno = :no",
            Employee.class)
            .setParameter("no", empno).getSingleResult();
        return emp;
    }
}
```

## Connection Pooling: UCP

```
package no.eritec.demo.SpringJpaDemo.config;

import java.sql.SQLException;
...

@Configuration
@Profile("oracle-ucp")
public class OracleUCPConfiguration {
    @Bean
    public DataSource dataSource() throws SQLException {
        PoolDataSource dataSource = PoolDataSourceFactory.getPoolDataSource();
        dataSource.setUser("appdata");
        dataSource.setPassword("app");
        dataSource.setConnectionFactoryClassName("oracle.jdbc.pool.OracleDataSource")
        dataSource.setURL("jdbc:oracle:thin:@//localhost:1521/ORCL");
        dataSource.setFastConnectionFailoverEnabled(true);
        dataSource.setInitialPoolSize(5);
        dataSource.setMinPoolSize(5);
        dataSource.setMaxPoolSize(5);
        return dataSource;
    }
}
```

**Create, manage, and run configurations**

Create a configuration that will launch a JUnit test.



type filter text

- Gradle Project
- Grunt
- Gulp
- Hibernate Console Configuration
- HTTP Preview
- J2EE Preview
- Java App In Container
- Java Applet
- Java Application
- JUnit
  - SpringBootDemoApplicationTests
  - SpringBootDemoApplicationTests.EmployeeJdbcRepos
  - SpringJpaDemoApplicationTests
- JUnit Plug-in Test
- Launch Group
- Maven Build
- Node.js Application
- OSGi Framework
- Pivotal tc Server
- Run Docker Image
- Spring Boot App
- Spring Boot Devtools Client
- Task Context Test
- XSL

Filter matched 34 of 87 items

Name: SpringJpaDemoApplicationTests

Test (x)= Arguments Classpath JRE Source Environment Common

Program arguments:

VM arguments:  
-ea -Dspring.profiles.active=oracle-ucp

Use the -XstartOnFirstThread argument when launching with SWT

Working directory:

Default: \${workspace\_loc:SpringJpaDemo}  
 Other: [Workspace...](#) [File System...](#) [Variables...](#)

[Revert](#) [Apply](#)

[Close](#)[Run](#)

# Agenda



1

Install & Setup



2

Key knowledge



3

The "real" knowledge

# Install & setup



1

Install & Setup

<http://www.json.no/oracle/?p=1668>

- [Part 1: Install/setup Oracle database \(in docker\)](#)
- [Part 2: Installing Java \(JDK\), Eclipse and Maven](#)
- [Part 3: Git, Oracle schemas and your first Java application](#)
- [Part 4: Your first JDBC Application](#)
- [Part 5: Spring-Boot, JdbcTemplate & DB migration \(Using FlywayDB\)](#)
- [Part 6: Spring-boot, JPA and Hibernate \(this\)](#)

# The “real” knowledge



3

The "real" knowledge

<https://github.com/lasjen/JavaForOracleDBAs>

- [Part 1: Install/setup Oracle database \(in docker\)](#)
- [Part 2: Installing Java \(JDK\), Eclipse and Maven](#)
- [Part 3: Git, Oracle schemas and your first Java application](#)
- [Part 4: Your first JDBC Application](#)
- [Part 5: Spring-Boot, JdbcTemplate & DB migration \(Using FlywayDB\)](#)
- [Part 6: Spring-boot, JPA and Hibernate \(this\)](#)



[lasse.jenssen@eritec.no](mailto:lasse.jenssen@eritec.no)

Twitter: @lasjen

<http://www.eritec.no>

**eritec**