application.properties

```
server.port=8088
#==============================================================
#Jpa hibernate
spring.jpa.hibernate.ddl-auto=update
spring.jpa.hibernate.show-sql=true
spring.jpa.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.jpa.show-sql = true
#==============================================================
#datasource
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://docker-mysql:3306/db1
spring.datasource.username=user
spring.datasource.password=password
#==============================================================


#datasource
#spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
#spring.datasource.url=jdbc:mysql://localhost:3306/db20
#spring.datasource.username=root
#spring.datasource.password=root
#==============================================================
```

Employee.java

```
package com.example.demo;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;


import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;
```

```
@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Employee {
@Id
@GeneratedValue(strategy = GenerationType.AUTO)
        private int empno;
        private String empname;
        private String empemail;
        private int age ;



}
```

EmployeeController.java

```
package com.example.demo;

import java.util.List;
import java.util.logging.Logger;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.servlet.ModelAndView;
```

```
@Controller
public class EmployeeController {

    @Autowired
    EmployeeDAO dao;
    Logger log=Logger.getAnonymousLogger();

    @RequestMapping("/")
        public ModelAndView displaydefaultpage(HttpServletRequest
request,HttpServletResponse resposne) {
            log.info("entered into the request mapping of /");
            ModelAndView mv=new ModelAndView();
            mv.setViewName("index.jsp");
            return mv;
        }

    @RequestMapping("/insert")
    public ModelAndView inserting(HttpServletRequest request,HttpServletResponse
resposne) {
        log.info("entered into the request mapping of /insert");
        ModelAndView mv=new ModelAndView();
        Employee emp=new Employee();
        emp.setEmpname(request.getParameter("name"));
        emp.setEmpemail(request.getParameter("email"));
        emp.setAge(Integer.parseInt(request.getParameter("age")));
        log.info("entered assigned values");
        Employee ee=dao.insert(emp);
        log.info("inserted");
        if(ee!=null) {
            mv.setViewName("success.jsp");
        }

        return mv;
    }


    @RequestMapping("/getall")
    public ModelAndView getal(HttpServletRequest request,HttpServletResponse resposne) {
        log.info("entered into the request mapping of /getall");
        ModelAndView mv=new ModelAndView();
        List<Employee> list=dao.getall();
        mv.setViewName("display.jsp");
        mv.addObject("list", list);
        return mv;
    }
    }
```

EmployeeDAO.java

```
package com.example.demo;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
@Service
public class EmployeeDAO {
@Autowired
EmployeeRepo repo;

//insertion
public Employee insert(Employee e) {
        return repo.save(e);
}

///retrival
public List<Employee> getall(){
        return repo.findAll();
}
}
```

EmployeeRepo.java

```
package com.example.demo;

import org.springframework.data.jpa.repository.JpaRepository;

public interface EmployeeRepo extends JpaRepository<Employee,Integer> {

}
```

Dockerfile

```
From openjdk:8
EXPOSE 8088
ADD target/Springwithdb-0.0.1-SNAPSHOT.war  Springwithdb-0.0.1-SNAPSHOT.war
ENTRYPOINT [ "java","-jar","/Springwithdb-0.0.1-SNAPSHOT.war" ]
```

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
https://maven.apache.org/xsd/maven-4.0.0.xsd">
        <modelVersion>4.0.0</modelVersion>
        <parent>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-parent</artifactId>
                <version>2.7.14</version>
                <relativePath/> <!-- lookup parent from repository -->
        </parent>
        <groupId>com.example</groupId>
        <artifactId>Springwithdb</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <packaging>war</packaging>
        <name>Springwithdb</name>
        <description>Demo project for Spring Boot</description>
        <properties>
                <java.version>1.8</java.version>
        </properties>
        <dependencies>
                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-data-jpa</artifactId>
                </dependency>
                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-web</artifactId>
                </dependency>

                <dependency>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-devtools</artifactId>
                        <scope>runtime</scope>
                        <optional>true</optional>
                </dependency>
                <dependency>
                        <groupId>com.mysql</groupId>
                        <artifactId>mysql-connector-j</artifactId>
                        <scope>runtime</scope>
                </dependency>
                <dependency>
                        <groupId>org.projectlombok</groupId>
                        <artifactId>lombok</artifactId>
```

```
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>

        <!-- JASPER DEPENDENCY -->
<dependency>
  <groupId>org.apache.tomcat</groupId>
  <artifactId>tomcat-jasper</artifactId>
  <version>9.0.78</version>
</dependency>

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <configuration>
                    <excludes>
                        <exclude>
                            <groupId>org.projectlombok</groupId>
                            <artifactId>lombok</artifactId>
                        </exclude>
                    </excludes>
                </configuration>
            </plugin>
        </plugins>
    </build>

</project>
```

Run docker

```
docker run --name docker-mysql -e MYSQL_ROOT_PASSWORD=password -e
MYSQL_DATABASE=db1 -e MYSQL_USER=user -e MYSQL_PASSWORD=password -d
mysql:latest --default-authentication-plugin=mysql_native_password
```

To start mysql image

```
docker exec -it docker-mysql mysql -u user -ppassword
```