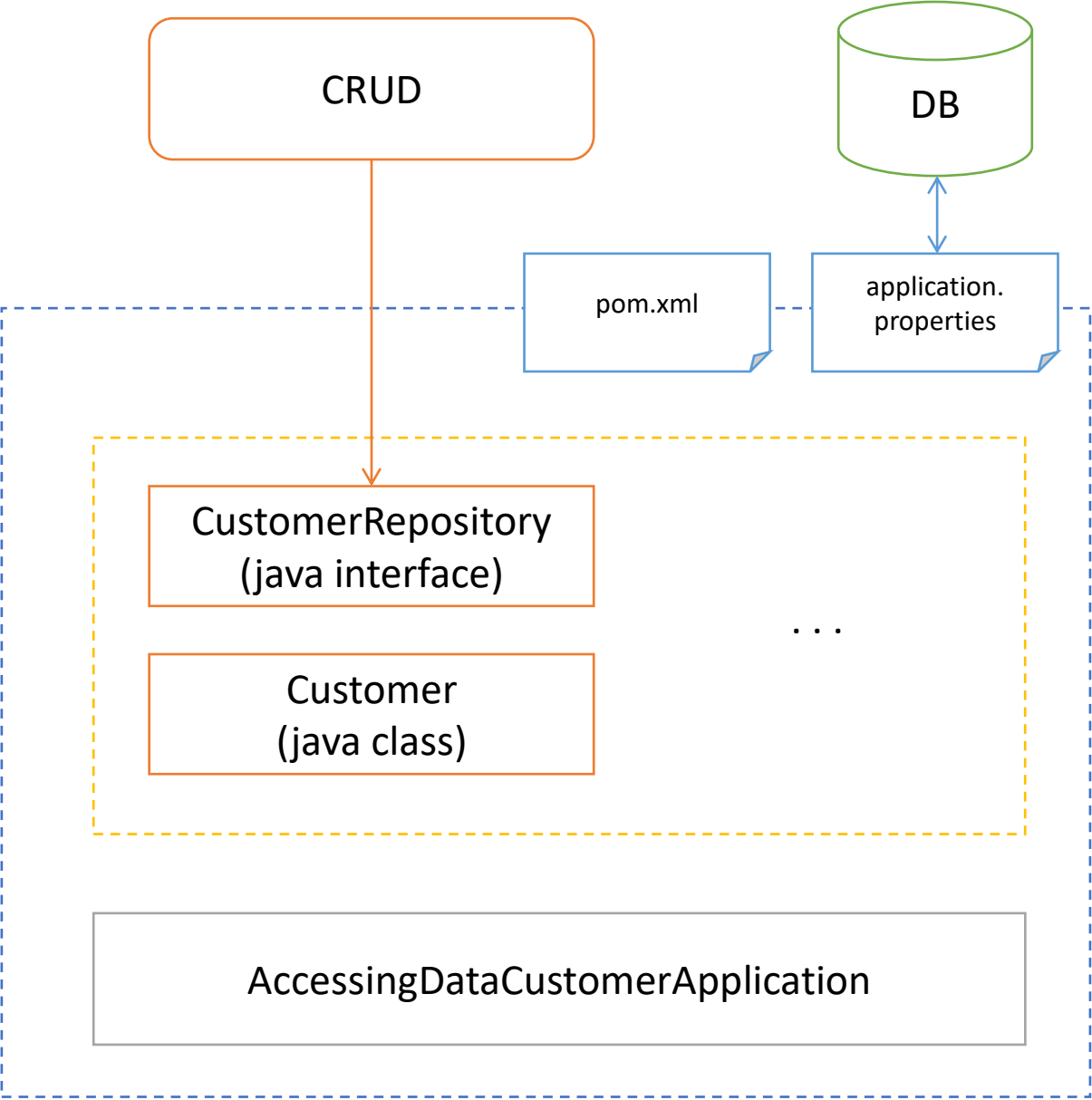


# Accessing Data with JPA

<https://spring.io/guides/gs/accessing-data-jpa/>

.idea  
src  
main  
java  
com.example.accessingdatajpa  
controllers  
...  
models  
Customer  
repositories  
CustomerRepository  
AccessingDataCustomerApplication  
resources  
application.properties  
test  
target  
pom.xml



# 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>
```

```
  <parent>
```

```
    <groupId>org.springframework.boot</groupId>
```

```
    <artifactId>spring-boot-starter-parent</artifactId>
```

```
    <version>2.2.2.RELEASE</version>
```

```
    <relativePath/> <!-- lookup parent from repository -->
```

```
  </parent>
```

```
  <groupId>com.example</groupId>
```

```
  <artifactId>accessing-data-jpa</artifactId>
```

```
  <version>0.0.1-SNAPSHOT</version>
```

```
  <name>accessing-data-jpa</name>
```

```
  <description>Demo project for Spring Boot</description>
```

```
  <dependencies>
```

```
    <dependency>
```

```
      <groupId>org.springframework.boot</groupId>
```

```
      <artifactId>spring-boot-starter-data-jpa</artifactId>
```

```
    </dependency>
```

```
      <dependency>
```

```
        <groupId>com.h2database</groupId>
```

```
        <artifactId>h2</artifactId>
```

```
        <scope>runtime</scope>
```

```
      </dependency>
```

```
</dependency>
```

```
  <groupId>org.springframework.boot</groupId>
```

```
  <artifactId>spring-boot-starter-test</artifactId>
```

```
  <scope>test</scope>
```

```
  <exclusions>
```

```
    <exclusion>
```

```
      <groupId>org.junit.vintage</groupId>
```

```
      <artifactId>junit-vintage-engine</artifactId>
```

```
    </exclusion>
```

```
  </exclusions>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>mysql</groupId>
```

```
  <artifactId>mysql-connector-java</artifactId>
```

```
  <scope>runtime</scope>
```

```
</dependency>
```

```
</dependencies>
```

```
<build>
```

```
  <plugins>
```

```
    <plugin>
```

```
      <groupId>org.springframework.boot</groupId>
```

```
      <artifactId>spring-boot-maven-plugin</artifactId>
```

```
    </plugin>
```

```
  </plugins>
```

```
</build>
```

```
</project>
```

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost:3306/db\_customer

spring.datasource.username=root

spring.datasource.password=

# Customer.java

```
package com.example.accessingdatajpa;

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

@Entity

public class Customer {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)

    private Long id;

    private String firstName;

    private String lastName;

    protected Customer() {}

    public Customer(String firstName, String lastName) {

        this.firstName = firstName;

        this.lastName = lastName;

    }
```

```
@Override

public String toString() {

    return String.format(

        "Customer[id=%d, firstName='%s', lastName='%s']",

        id, firstName, lastName);

}

public Long getId() {

    return id;

}

public String getFirstName() {

    return firstName;

}

public String getLastName() {

    return lastName;

}

}
```

Long	id	Auto Increment PC
String	firstName	
String	lastName	

```
package com.example.accessingdatajpa;
import java.util.List;
import org.springframework.data.repository.CrudRepository;
public interface CustomerRepository extends CrudRepository<Customer, Long> {
    List<Customer> findByLastName(String lastName);
    Customer findById(long id);
}
```

# AccessingDataJpaApplication.java

```
AccessingDataCustomerApplication : Customers found with findAll():
AccessingDataCustomerApplication : -----
AccessingDataCustomerApplication : Customer[id=6, firstName='Jack', lastName='Bauer']
AccessingDataCustomerApplication : Customer[id=7, firstName='Chloe', lastName='O'Brian']
AccessingDataCustomerApplication : Customer[id=8, firstName='Kim', lastName='Bauer']
AccessingDataCustomerApplication : Customer[id=9, firstName='David', lastName='Palmer']
AccessingDataCustomerApplication : Customer[id=10, firstName='Michelle',

AccessingDataCustomerApplication : Customer[id=11, firstName='Jack', lastName='Bauer']
AccessingDataCustomerApplication : Customer[id=12, firstName='Chloe', lastName='O'Brian']
AccessingDataCustomerApplication : Customer[id=13, firstName='Kim', lastName='Bauer']
AccessingDataCustomerApplication : Customer[id=14, firstName='David', lastName='Palmer']
AccessingDataCustomerApplication : Customer[id=15, firstName='Michelle',

AccessingDataCustomerApplication :
AccessingDataCustomerApplication : Customer found with findById(10L):
AccessingDataCustomerApplication : -----
AccessingDataCustomerApplication : Customer[id=10, firstName='Michelle',

AccessingDataCustomerApplication :
AccessingDataCustomerApplication : Customer found with findByLastName('Bauer'):
AccessingDataCustomerApplication : -----
AccessingDataCustomerApplication : Customer[id=6, firstName='Jack', lastName='Bauer']
AccessingDataCustomerApplication : Customer[id=8, firstName='Kim', lastName='Bauer']
AccessingDataCustomerApplication : Customer[id=11, firstName='Jack', lastName='Bauer']
AccessingDataCustomerApplication : Customer[id=13, firstName='Kim', lastName='Bauer']
```

```
repository.save(new Customer("Jack", "Bauer"));

repository.save(new Customer("Chloe", "O'Brian"));

repository.save(new Customer("Kim", "Bauer"));

repository.save(new Customer("David", "Palmer"));

repository.save(new Customer("Michelle", "Dessler"));
```

id	firstName	lastName
1	Jack	Brauer
2	Chloe	O'Brian

```
// fetch all customers

log.info("Customers found with findAll():");

log.info("-----");

for (Customer customer : repository.findAll()) {

    log.info(customer.toString());

}

log.info("");

// fetch an individual customer by ID

Customer customer = repository.findById(10L);

log.info("Customer found with findById(10L):");

log.info("-----");

log.info(customer.toString());

log.info("");

// fetch customers by last name

log.info("Customer found with findByLastName('Bauer'):");

log.info("-----");

repository.findByLastName("Bauer").forEach(bauer -> {

    log.info(bauer.toString());

});

// for (Customer bauer : repository.findByLastName("Bauer")) {

// log.info(bauer.toString());

// }

log.info("");

};

}

}
```