

## Overview



Data access

Benefits of Spring with JPA

Default configurations

Querying using repositories



## H2 Database

---



# H2 Database



Open source database written in Java



In-memory database



Good for POCs, dev environments, simple database



Administer via the H2 console



```
<dependency>
```

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

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

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

```
</dependency>
```

---

## H2 Dependency

**Auto-configures H2 related properties**



## H2 Defaults

```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.h2.console.enabled=false
```



## Override Defaults

```
spring.h2.console.enabled=true
```

```
spring.h2.console.path=/h2
```

```
spring.datasource.url=jdbc:h2:mem:bugtracker
```

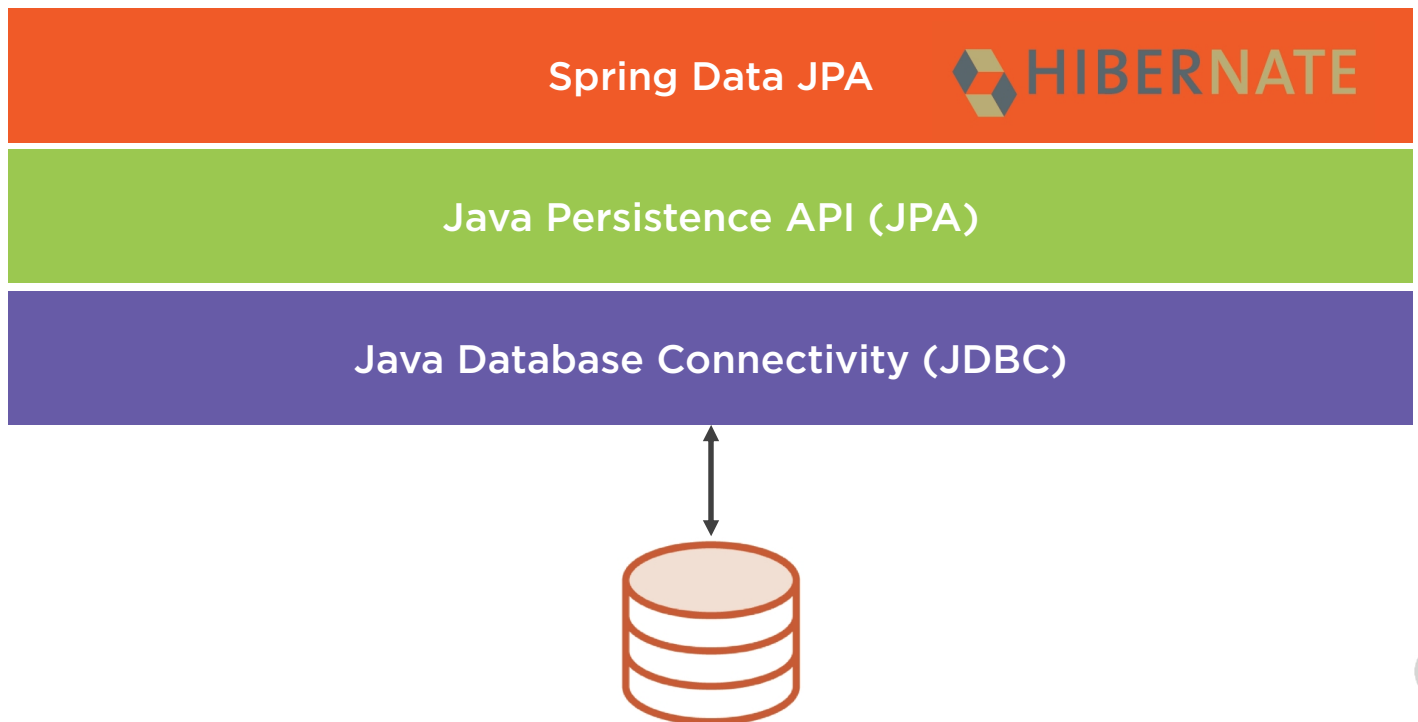


## ORM with JPA

---



## ORM with JPA





```
<dependency>
```

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

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

```
</dependency>
```

---

## Spring Data

Hibernate, Spring Data JPA, Spring ORM



# Entities

---



## Entities

```
package com.pluralsight.entity;

import javax.persistence.*;

@Entity

public class Application {

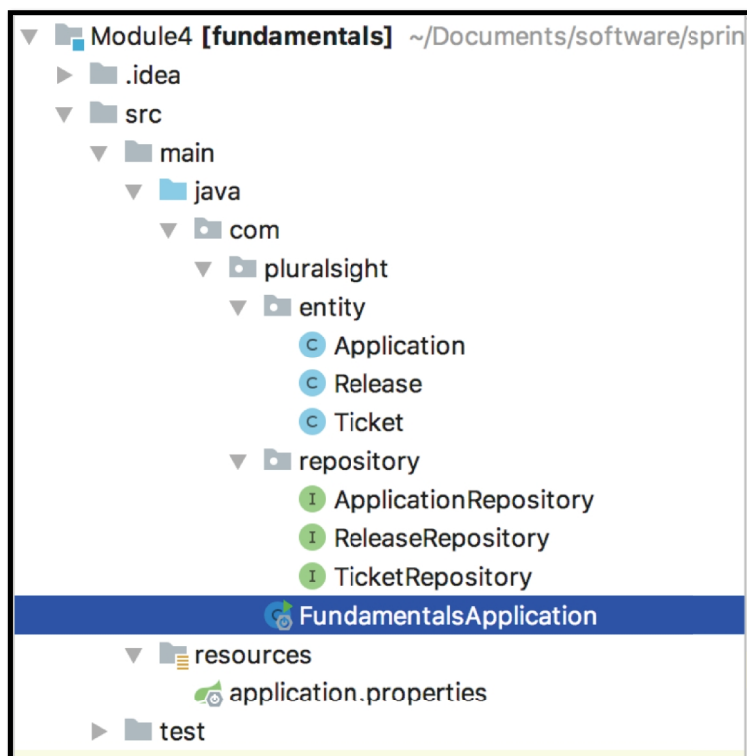
    @Id

    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name="application_id")
    private Integer id;

    @Column(name = "app_name", nullable = false)
    private String name;
```



# Entity Scanning



Demo



H2 database access

Spring Data JPA

Entity Scanning

Repositories



## Summary



**H2 data access using Spring Boot**

**ORM with JPA**

**Benefits of Spring with JPA**

**Entities**

**Simple queries with Repositories**

