ORM support in Spring

Agenda

- Selection of ORM
- JPA Practical Example
  - Creation of EntityManager bean
  - Code

Selection of ORM provider

- JPA API
  - Spring comes with Hibernate as default JPA provider
- Hibernate API
- In our practical example JPA will be used

Gradle dependency for Boot project

- compile('**org.springframework.boot:spring-boot-starter-data-jpa**')

Create JPA EntityManager

- Programmatic configuration
- Automatic configuration

Java Configuration – see example

```
@Configuration
@EnableTransactionManagement
public class Lesson4Configuration  {
   @Autowired  private DataSource datasource;

   @Bean
   public LocalContainerEntityManagerFactoryBean entityManagerFactory() {
      LocalContainerEntityManagerFactoryBean em = new LocalContainerEntityManagerFactoryBean();
      em.setDataSource(datasource);
      em.setPackagesToScan(new String[] { "lesson4.model" });
      em.setJpaVendorAdapter(new  HibernateJpaVendorAdapter());
      em.setJpaProperties(…);
      return em;
   }

   @Bean
   public PlatformTransactionManager transactionManager(EntityManagerFactory emf){
      JpaTransactionManager transactionManager = new JpaTransactionManager();
      transactionManager.setEntityManagerFactory(emf);
      return transactionManager;
   }
```

Automatic configuration

- @EnableAutoConfiguration will let Boot automatically configure an DataSource, a LocalContainerEntityManagerFactoryBean for Hibernate as well as a JPA transaction manager for JPA.
- spring.jpa.* doc https://docs.spring.io/spring-boot/docs/current/reference/html/common-application-properties.html

application.properties in OUR EXAMPE

```
spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.show-sql=true
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
```

Entities & Daos code

- See example