

```

package br.com.alura.owasp.infra;

import java.util.Properties;

import javax.persistence.EntityManagerFactory;
import javax.sql.DataSource;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.env.Environment;
import org.springframework.dao.annotation.PersistenceExceptionTranslationPostProcessor;
import org.springframework.jdbc.datasource.DriverManagerDataSource;
import org.springframework.orm.jpa.JpaTransactionManager;
import org.springframework.orm.jpa.JpaVendorAdapter;
import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;
import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;
import org.springframework.transaction.PlatformTransactionManager;
import org.springframework.transaction.annotation.EnableTransactionManagement;

```

```
@Configuration
```

```
@EnableTransactionManagement
```

```
public class JPAConfiguration {
```

```
    @Bean
```

```
        public LocalContainerEntityManagerFactoryBean
entityManagerFactory(DataSource dataSource) {
            LocalContainerEntityManagerFactoryBean em = new
LocalContainerEntityManagerFactoryBean();
            em.setDataSource(dataSource);
            em.setPackagesToScan(new String[] { "br.com.alura.owasp.model" });

            JpaVendorAdapter vendorAdapter = new HibernateJpaVendorAdapter();
            em.setJpaVendorAdapter(vendorAdapter);
            em.setJpaProperties(additionalProperties());

            return em;
        }

```

```
    @Bean
```

```
        public DataSource dataSource(Environment environment){
            DriverManagerDataSource dataSource = new DriverManagerDataSource();
            dataSource.setDriverClassName("com.mysql.jdbc.Driver");

            dataSource.setUrl("jdbc:mysql://localhost:3306/owasp?createDatabaseIfNotExist=true");
            dataSource.setUsername( "root" );
            dataSource.setPassword( "" );
        }

```

```

        return dataSource;
    }

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

    @Bean
    public PersistenceExceptionTranslationPostProcessor exceptionTranslation(){
        return new PersistenceExceptionTranslationPostProcessor();
    }

    Properties additionalProperties() {
        Properties properties = new Properties();
        properties.setProperty("hibernate.dialect",
"org.hibernate.dialect.MySQL5Dialect");
        properties.setProperty("hibernate.hbm2ddl.auto", "update");
        properties.setProperty("hibernate.show_sql", "true");
        return properties;
    }
}

```