

Spring Data JPA

Arnaud Cogoluègnes

Zenika

March 1, 2012

Plan

Spring JPA

Spring Data JPA Hello World

Plan

Spring JPA

Spring Data JPA Hello World

LocalContainerEntityManagerFactoryBean

- ▶ Pour la configuration de JPA dans Spring
- ▶ Fait partie du Spring Framework (module ORM)
- ▶ Flexible :
 - ▶ Injection de dépendances (ex. : DataSource)
 - ▶ Compatible avec les différentes implémentations JPA
 - ▶ Pas besoin de `persistence.xml`
 - ▶ Scanning de packages pour trouver les entités

Configuration de JPA avec Spring

```
<bean id="entityManagerFactory"
      class="o.s.orm.jpa.LocalContainerEntityManagerFactoryBean">
  <property name="dataSource" ref="dataSource" />
  <property name="jpaVendorAdapter">
    <bean class="o.s.orm.jpa.vendor.HibernateJpaVendorAdapter">
      <property name="showSql" value="true" />
      <property name="generateDdl" value="true" />
      <property name="database" value="H2" />
    </bean>
  <property name="packagesToScan" value="com.zenika.nordnet.model" />
</bean>
```

Entité JPA

```
package com.zenika.nordnet.model;

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

@Entity
public class Contact {

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE)
    private Long id;

    private String firstname, lastname;

    (...)
}
```

TestContext framework

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("/spring-jpa.xml")
public class SpringJpaTest {

    @Autowired private EntityManagerFactory emf;

    @Test public void springJpa() {
        (...)
    }
}
```

Méthode de test

```
@Test public void springJpa() {  
    EntityManager em = emf.createEntityManager();  
    em.getTransaction().begin();  
    try {  
        Contact contact = new Contact();  
        contact.setFirstname("Mickey");  
        contact.setLastname("Mouse");  
        int initialCount = em.createQuery("from Contact")  
            .getResultList().size();  
        em.persist(contact);  
        Assert.assertEquals(  
            initialCount+1,  
            em.createQuery("from Contact").getResultList().size()  
        );  
        em.getTransaction().commit();  
    } finally {  
        em.close();  
    }  
}
```


Configuration Java (alternative)

```
@Configuration
public class SpringJpaConfiguration {

    @Bean public EntityManagerFactory emf() {
        LocalContainerEntityManagerFactoryBean emf =
            new LocalContainerEntityManagerFactoryBean();
        emf.setDataSource(ds());
        HibernateJpaVendorAdapter adapter = new HibernateJpaVendorAdapter();
        adapter.setGenerateDdl(true);
        emf.setJpaVendorAdapter(adapter);
        emf.setPackagesToScan(Contact.class.getPackage().getName());
        emf.afterPropertiesSet();
        return emf.getObject();
    }

    @Bean public DataSource ds() {
        return new EmbeddedDatabaseBuilder()
            .setType(EmbeddedDatabaseType.H2)
            .build();
    }
}
```

Configuration Java dans le test

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(classes=SpringJpaConfiguration.class)
public class SpringJpaJavaConfigurationTest {

    @Autowired private EntityManagerFactory emf;

    @Test public void springJpa() {
        (...)
    }
}
```

Plan

Spring JPA

Spring Data JPA Hello World

Interface de repository

- ▶ Utiliser JpaRepository comme interface de base
- ▶ Typer l'interface avec la classe d'entité et la classe de l'identifiant

```
package com.zenika.nordnet.repository;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import com.zenika.nordnet.model.Contact;  
  
public interface ContactRepository  
    extends JpaRepository<Contact, Long> {  
  
}
```

Déclarer les repositories

- ▶ Indiquer à Spring Data JPA les packages des repositories
- ▶ Spring détecte les repositories et crée les beans correspondants

```
<jpa:repositories base-package="com.zenika.nordnet.repository" />
```

- ▶ NB : utilisation du namespace jpa

Infrastructure

- ▶ Dépendances de `<jpa:repositories />` :
 - ▶ `EntityManagerFactory`
 - ▶ `PlatformTransactionManager`

```
<bean id="entityManagerFactory"
      class="o.s.orm.jpa.LocalContainerEntityManagerFactoryBean">
  (...)
</bean>

<bean id="transactionManager"
      class="org.springframework.orm.jpa.JpaTransactionManager">
  <property name="entityManagerFactory" ref="entityManagerFactory" />
</bean>
```

Test !

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("/spring-data-jpa-hello-world.xml")
public class SpringDataJpaHelloWorldTest {

    @Autowired ContactRepository repo;

    @Test public void springJpa() {
        Contact contact = new Contact();
        contact.setFirstname("Mickey");
        contact.setLastname("Mouse");
        long initialCount = repo.count();
        repo.save(contact);
        Assert.assertEquals(initialCount + 1, repo.count());
    }
}
```