

Hibernate

Piotr Mazur

Katedra Mikroelektroniki i Technik Informatycznych

Łódź, 3 listopada 2010

- 1 Wprowadzenie
 - Informacje organizacyjne

Wprowadzenie

Hibernate jest biblioteką służącą do **mapowania obiektowo - relacyjnego**

Mapowanie pozwala na odwzorowanie struktury klas wraz z ich powiązaniem w postaci schematu bazy danych

Uzytkownik.java

```
public class Uzytkownik {  
    private Integer id;  
    private String username;  
    private String password;  
  
    /* metody get i set */  
}
```

Tabela uzytkownik

Nazwa kolumny	id	username	password
Wiersz 1	3	Jan	Kowalski
Wiersz 2	5	Kasia	Strzelecka

Typy obiektów

Hibernate dzieli obiekty na kilka podstawowych typów

- **Transient** - obiekt nie powiązany z bazą danych
- **Persistent** - obiekt powiązany z bazą danych, sesja dalej aktywna
- **Detached** - obiekt powiązany z bazą danych, bez aktywnej sesji

Ważniejsze anotacje

@Entity - Oznaczenie klasy jako encji, encja jest modelem do odwzorowania w bazie danych

@Id - Oznaczenie pola jako klucza głównego. Za jego pomocą identyfikowany jest dany obiekt. W bazie danych zostanie automatycznie utworzona kolumna z kluczem głównym

@OneToMany - Jedna z końcówek relacji jeden-do-wielu. (*część z listą*)

@ManyToOne - Drugi koniec relacji typu **jeden-do-wielu**. (*część bez listy*)

Przykład klasy

Uzytkownik.java

```
@Entity
public class Uzytkownik {

    @Id
    @GeneratedValue
    private Integer id;

    private String username;

    private String password;

    /* metody get i set */
}
```

Mapowanie jeden do wielu

Autor.java

```
@Entity
public class Autor {
    @Id
    @GeneratedValue
    private Integer id;
    private String name;
    private String surname;

    @OneToMany(mappedBy="autor")
    private List<Ksiazka> ksiazki;

    /* metody get i set */
}
```

Ksiazka.java

```
@Entity
public class Ksiazka {
    @Id
    @GeneratedValue
    private Integer id;
    private String title;

    @ManyToOne
    private Autor autor;

    /* metody get i set */
}
```


SessionFactory

Przykład wykonania operacji zapisu obiektu do bazy danych.
Zakłada istnienie beana o nazwie "sessionFactory" w kontekście aplikacji (Spring)

```
SessionFactory factory=context.getBean("sessionFactory");  
Session session=factory.openSession();  
Transaction t=session.beginTransaction();  
session.save(uzytkownik);  
t.commit();  
session.close();
```

```
<bean id="sessionFactory"
  class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean" >
  <property name="packagesToScan" >
    <list>
      <value>org.dmcs.zai.*</value>
    </list>
  </property>

  <property name="hibernateProperties">
    <props>
      <prop key="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</prop>
      <prop key="hibernate.query.substitutions">>true=1 false=0</prop>
      <prop key="hibernate.show_sql">>true</prop>
      <prop key="hibernate.format_sql">>true</prop>
    </props>
  </property>
  <property name="dataSource"><ref bean="dataSource"/></property>
  <property name="schemaUpdate"><value>true</value></property>
</bean>
```